Filtrado de Spam mediante Ajuste Lineal por Cuadrados Mínimos

Tesistas

Pablo Alejando Alvarez Alonso
LU 503/87
pablo@sisbi.uba.ar

Daniel Mario Vega
LU 195/88
dv5a@dc.uba.ar

Directora

Dra. Ana Silvia Haedo
haedo@qb.fcen.uba.ar

Resumen

Un problema creciente en las comunicaciones mediante correo electrónico es la práctica de utilizar este medio para el envío de mensajes publicitarios masivos no solicitados, mejor conocidos como “Spam”. Distintas soluciones han sido propuestas para atacar este problema, como ser la utilización de técnicas de aprendizaje automático.

En este trabajo de tesis, analizaremos un método de clasificación y filtrado basado en ajuste lineal por cuadrados mínimos (LLSF) [YAN/94] en la tarea de filtrado de Spam. Analizaremos distintas variantes y mejoras sobre el algoritmo básico. Entre ellas presentaremos una nueva fórmula de selección de atributos, nuevas alternativas en la representación de los mensajes, un método matemático de determinación del umbral. Finalmente comparemos los resultados con los obtenidos en trabajos anteriores, los cuales utilizaron el algoritmo de Naïve-Bayes [AND/00b].

1. Introducción

Toda persona que haya utilizado el correo electrónico por cierto tiempo habrá seguramente sido objeto de mensajes publicitarios masivos no solicitados, mejor conocidos como correo basura o simplemente como "Spam". Esta técnica publicitaria tiene una serie de características que la vuelven indeseable.

1.1 ¿ Qué hay de malo con el Spam ?

> *"El correo basura es el flagelo del correo electrónico y los grupos de noticias en Internet. Puede interferir seriamente con la operación de servicios públicos, por no mencionar el efecto que puede tener en los sistemas de correo electrónico de cualquier individuo... Los spammers están, de forma efectiva, sustrayendo recursos de los usuarios y proveedores de servicio sin compensación y sin autorización."*
>
> Vint Cerf, Vicepresidente de MCI
> conocido como "Padre de Internet"

Es común que recibamos publicidad mediante el servicio de correo postal. Nos llega desde el Banco donde tenemos cuenta, la tarjeta de crédito, o el local de electrodomésticos donde solemos comprar. Es algo normal y en su mayor parte no nos molesta. ¿ Entonces porque la publicidad por correo electrónico es distinta?.

El "Spam" tiene una serie de características que lo diferencian de otros tipos de herramientas publicitarias, las cuales lo vuelven indeseable:

- **Trasferencia de costos**. La mayor parte del costo es pagado por el receptor en vez del emisor.

  Es muy sencillo y rápido enviar estos mensajes. Con un modem y una línea telefónica en teoría es posible transmitir cientos de miles de estos mensajes por hora[1]. En cambio cada uno de los destinatarios debe mal-

[1] Para enviar un mensaje a múltiples recipientes utilizando el protocolo SMTP es necesario transmitir la lista de destinatarios (direcciones de correo electrónico) y luego el contenido del mensaje. Por lo que agregar un nuevo destinatario sólo incrementa en unos pocos bytes el volumen de datos a ser transferido.

gastar parte de su tiempo para descargar y clasificar estos mensajes. Esto se traduce en mayores costos de conexión[2]. Un estudio patrocinado por la Comunidad Económica Europea estimó que los usuarios de Internet de todo el mundo están pagando diez mil millones de Euros por año, solo considerando costos de conexión, por culpa de estos mensajes publicitarios[3].

- **Uso de engaños**. Los emisores de este tipo de mensajes (de ahora en más *Spammers*) saben que los destinatarios en su gran mayoría no desean recibirlos, por lo que hacen uso de distintas artimañas para inducirlos a abrir el mensaje. Por ejemplo colocando un "tema" en el mensaje tal que no parezca que se trata de una publicidad. Otro truco común es el de utilizar una dirección de origen falsa[4]. También suelen insertar encabezados falsos en el mensaje para ocultar su origen y posible detección.

- **Utilización de Recursos Ajenos**. Para poder enviar sus millones de mensajes, los spammers utilizan recursos de CPU, ancho de banda, y espacio en disco de cada uno de los equipos por los que pasa cada uno de ellos. Los portadores de dichos mensajes deben hacerse cargo del costo de publicidad de los spammers.

  Muchas actividades económicas suelen producir molestias a terceros. Por ejemplo una fabrica puede emitir ruidos molestos, una locomotora despide humo, una represa afecta al medio ambiente. Pero para poder determinar si una actividad es beneficiosa o no, hay que ponderar la riqueza que genera dicha actividad versus los costos sociales de la misma. Bajo condiciones ideales de mercado, las actividades no lucrativas tenderían a desaparecer.

---

[2] Esto se refleja, por ejemplo, en mayores facturas de teléfono. Y aunque el usuario tenga una conexión con tarifa plana (o sea que paga una suma fija de dinero por mes), esta tarifa se calcula en base a los gastos que el proveedor debe asumir para ofrecer el servicio (ancho de banda, CPU, espacio de almacenamiento en discos, etc.), que se ven afectados por este tipo de mensajes.

[3] Commission of the European Communities, 2002, "Junk e-mail costs internet users euro 10 billion a year worldwide",
http://europa.eu.int/comm/internal_market/en/dataprot/studies/spam.htm

[4] Los spammers no utilizan como dirección de origen de sus mensajes su verdadera dirección de email. Al utilizar una dirección falsa, se vuelve más difícil el poder rastrear el origen o elevar una queja. Por otro lado al utilizar siempre direcciones distintas se dificulta el poder "bloquear" sus mensajes. Y por último como las bases de direcciones de correo que utilizan para enviar mensajes suelen tener errores o estar desactualizadas, evitan tener que lidiar con los numerosos avisos de "correo no entregado" que se generan.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Si un individuo **A** para obtener una ganancia de $2 perjudica a otra persona **B** por $100, ésta (de no tener alternativa más económica) estaría dispuesta pagarle $2 (o incluso más) a **A** para que deje de hacer lo que esta haciendo[COA/60]. Pero normalmente llegar a estos acuerdos también tiene un costo, por lo que si un individuo **A** para obtener una ganancia de $2 afecta a 10.000 personas en $0,01 cada una, estas posiblemente aceptarán el costo ya que organizarse para detener dicha actividad es más costoso que no hacer nada[5].

Los spammers saben que molestando a millones de personas sólo un "poquito", posiblemente puedan seguir adelante con su negocio.

- **Pérdida de Mensajes Legítimos**. Al incrementarse el número de Spams que son enviados su efecto sobre el correo legítimo aumenta de igual modo. Si nuestro buzón de entrada tiene muchos mensajes, es posible que clasifiquemos por error como Spam un mensaje que no lo es. Especialmente si el remitente no nos resulta conocido. Otro problema que se presenta es que si pasamos algunos días sin revisar nuestro correo, puede suceder que el buzón de entrada agote su capacidad y correo legitimo no pueda ser procesado.

Los spammers usan diversos métodos para recolectar sus bases de direcciones de correo electrónico. Algunas de las formas más empleadas son tomándolas de páginas web donde fueron publicadas, copiándolas de los mensajes enviados a grupos de discusión o listas de correo públicas, o comprándolas (o intercambiándolas) a sitios web en los que uno tiene que ingresar su dirección de email, entre otras[6].

[5] Normalmente es el estado el que interviene en estos casos prohibiendo o regulando la actividad que es perjudicial para la sociedad. Por ejemplo cuando se le prohibe a una empresa arrojar desperdicios a un lago.

[6] Para más información ver "Why Am I Getting All This Spam?" Report on Origins of Spam. Center for Democracy & Technology. March 2003
http://www.cdt.org/speech/spam/030319spamreport.pdf

1.2 Breve Historia del Spam

La historia de los mensajes comerciales de correo electrónico comienza a fines de la década del 70[7] y llega a nuestros días. Se propusieron distintos nombres para referirse a este tipo de mensajes, "UBE" (*Unsolicited Bulk Email*), "UCE" (*Unsolicited Commercial Email*), "Correo Basura" (*Junk Mail*), entre otros. Pero actualmente se los suele denominar simplemente "Spam". El camino que recorrió esta palabra para llegar a tener este significado es por lo menos "curioso".

1.2.1 El origen de la palabra "Spam"

"SPAM SPAM SPAM SPAM
Hormel's new miracle meat in a can
Tastes fine, saves time.
If you want something grand,
Ask for SPAM!"

Comercial Radial de 1940

En 1926 la empresa alimenticia Hormel Foods comenzó a fabricar en EEUU carne de cerdo enlatada. El producto fue exitoso (principalmente en los años de la gran depresión) por ser nutritivo y económico. Diez años después lanzaron un nuevo producto que consistía en carne de cerdo cocida, sazonada y enlatada, que no requería de refrigeración para su mantenimiento. El producto se llamó *Hormel Spiced Ham* (Jamón Sazonado Hormel), pero al año siguiente decidieron buscarle un nuevo nombre que fuera distintivo y sirviese para diferenciarlo de la competencia. El nombre elegido fue "SPAM", que surgió de la contracción de *Spiced Ham* (Jamón Sazonado)[8].

[7] El Spam más antiguo que se tiene registro es del 1 de Mayo de 1978. Fue enviado en la red ARPANET por un empleado de DEC (Digital Equipment Corporation) anunciando la presentación de nuevos modelos de computadoras.

[8] El SPAM tiene una apariencia de paleta de cerdo enlatada. Se lo considera como una alternativa barata a la carne y de menor calidad que el jamón. Es por eso que se pueden encontrar en algunos estudios de técnicas y herramientas para combatir *Spam* (mensajes publicitarios), en que a los mensajes legítimos se los denomina *Ham* (Jamón).

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Este producto tuvo un gran auge durante la segunda guerra mundial, se lo utilizó para alimentar a las tropas, se exportó al resto de los países aliados, y en EEUU su uso creció rápidamente dado que el consumo de carne vacuna había sido racionado.

"Sin SPAM no hubiésemos podido alimentar a nuestro ejército"
— Nikita Kruschev, Líder Soviético

Desde su introducción en el mercado se han producido más de 5 mil millones de latas de 200 y 340 gramos de este producto.

1.2.2 Carne Enlatada y Correo Electrónico Basura

"Shhh, querida, no hagas escándalo. Yo tomaré tu spam. ¡Me encanta! Pediré spam, spam, spam, spam, spam, spam, spam, frijoles, spam, spam, spam y spam"

— Escena de comedia del "Circo Volador de Monty Python"

Para explicar como llega el término Spam a su actual significado hay que hacer un poco de historia.

A comienzos de la década del 70 la televisión Inglesa emitía al aire un programa cómico llamado "El Circo Volador de Monty Python". En uno de sus episodios se presentó una escena en que un matrimonio ingresa a un restaurante donde todos los platos contenían mucho Spam. La camarera repetía continuamente esta palabra al recitar el menú. En otra mesa un grupo de Vikingos comienza a entonar una canción de alabanza al Spam que impiden que el matrimonio y la camarera puedan seguir conversando.

"Spam, spam, spam, spam, spam, spam. ¡ Spam precioso ! ¡ Spam maravilloso !"

A mediados de los 80s surge una red conversación llamada *BITNET Relay*. El propósito de la misma era crear una red para conferencias académicas. Este fue el antecesor de lo que después se convirtió en IRC (Internet Relay Chat). Este sistema consistía básicamente en una serie de salas donde los participantes podían charlar entre sí. Según cuentan los memoriosos a veces algún participante se ponía a repetir la canción de alabanza al spam con el solo hecho de molestar a los demás y no permitir la conversación normal[9]. Esta actividad de enviar múltiples mensajes en rápida sucesión con el objeto de molestar a otros usuarios o sobrecargar el sistema se llamó "hacer spam".

De allí la palabra Spam pasó a la red de grupos noticias[10] *USENET*. Ésta es una red de mensajes que surgió a fines de la década del 70 donde los participantes pueden enviar y leer mensajes de distintos grupos[11]. Estos mensajes son replicados entre los servidores que participan de la red.

A comienzos de los 90s ya habían ocurrido algunos casos de envíos de publicidad a distintos grupos de dicha red, pero ninguno de ellos fue masivo ni tampoco llamado Spam. Hasta que en marzo de 1993 mientras uno de los operadores de la red estaba probando un programa nuevo de "moderación" de los grupos, éste envió por accidente una serie de mensajes sucesivos (alrededor de 200) a uno de los grupos. A este incidente se lo denominó por primera vez como un envío de *Spam*.

En abril de 1994 un estudio de abogados llamado "Canter & Siegel" contrató un programador para que enviara a cada uno de los grupos de debate un mensaje donde publicitaban los servicios de asesoría para obtener visas de residencia en EEUU. Esta lluvia de mensajes causó muchas protestas e indignación entre los usuarios. Pero lamentablemente el envío de publicidad masiva se fue haciendo cada vez más común. De allí la práctica pasó al correo electrónico donde heredó el apelativo de Spam.

[9] Este comportamiento "vandálico" fue también presenciado en BBSs y MUDs (Multi-User Domain. Salas de juego multiusuarios en línea), por lo que hay alguna discusión acerca de dónde se originó.

[10] *Discussion Groups*. También conocida como *Newsgroups*. Grupos de Noticias.

[11] Los grupos se dividen de acuerdo a temas de interés.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

1.3 Métodos para combatir el Spam

Dada la impopularidad de estos tipos de mensajes, se han estudiado e implementado una variedad de técnicas para combatirlos.

Notificar al Proveedor del Servicio de Internet (PSI)

Muchos PSIs reconocen los inconvenientes que generan este tipo de mensajes y tienen en sus contratos de servicio "Políticas de Uso Aceptable" (PUA), que establecen que sus usuarios no pueden enviar Spam. Por lo que al ser notificados[12] de algún incumplimiento por parte de sus clientes estos suelen implementar sanciones que van desde un apercibimiento hasta la cancelación del servicio de Internet.

Este mecanismo para combatir el Spam es muy útil y ha ayudado a contener en parte el problema. Pero lamentablemente es insuficiente. Los spammers simplemente cambian de proveedor de Internet, o utilizan algún proveedor gratuito, o incluso utilizan cuentas de PSIs que fueron robadas de usuarios inocentes[13]. Además existen algunos proveedores de Internet a los que simplemente no les importa que sus usuarios envíen este tipo de mensajes.

Listas Negras

Dado que algunos proveedores de Internet no realizan políticas activas de lucha contra el Spam, algunos grupos han decidido implementar *listas negras*. Esto es, organizan bases de datos de direcciones IP a las que consideran que suelen ser fuente de Spam y configuran sus servidores de email para que rechacen cualquier mensaje que provenga de estas direcciones prohibidas.

Algunos años atrás todos los servidores de correo estaban configurados de la forma que actualmente se conoce como *OpenRelay*. Esto es que los servidores actuaban como si fuesen oficinas postales, y todos ellos aceptaban correo proveniente de cualquier persona y dirigido hacia cualquier otra. Pero algunos

[12] La mayoría de los PSIs tienen una dirección de email a la cual se pueden denunciar estas faltas. Generalmente con la forma *abuse@provedor.com*. Existe además en Internet una base de datos de direcciones de email a la cual denunciar estos incidentes. http://www.abuse.net/lookup.phtml .

[13] Una forma común de robo de cuentas es enviando un email a una gran cantidad de usuarios de algún PSI importante donde aparentan ser del departamento de soporte técnico y le solicitan al usuario (arguyendo alguna reestructuración del sistema) que por favor se dirija a una dirección web y reingresen sus datos (usuario y contraseña).

administradores notaron que los Spammers estaban consumiendo los recursos de sus servidores y comenzaron a restringir el uso de los mismos. Sólo aceptaban correo si el origen o el destino pertenecían a su subred[14].

Las primeras listas negras tuvieron como objetivo concientizar a los administradores para que modifiquen la configuración de sus servidores. Un servidor era incluido en la lista negra si se trataba de un *OpenRelay* y luego de notificar al administrador éste no hacía nada para solucionarlo.

Hoy en día existen muchas de estas listas[15], y los criterios para ser incluido en ellas son bastante variados. Existen algunas listas más restrictivas que otras.

La creación y uso de estas listas han sido polémicos. Si bien han ayudado a combatir el Spam y muchos PSIs tomaron conciencia del problema al ser sus servidores incluidos en estas *Listas Negras*, también es cierto que afectan a gente inocente y bloquean la transferencia de muchos mensajes legítimos[16].

Listas Blancas

Este sistema consiste en que cada usuario defina una lista de direcciones de email que están autorizadas a enviarles mensajes. Cualquier otro mensaje que provenga de una dirección no incluida en la lista blanca es descartado.

Este método si bien es muy restrictivo puede ser de utilidad en algunos casos, por ejemplo si una empresa tiene una cuenta de email mediante la cual se comunica con sus proveedores, puede implementar este sistema ya que conoce de antemano quienes son sus proveedores.

Existen algunos programas que implementan un método de actualización automática de la lista blanca. El método se basa en que la dirección origen de los mensajes Spam no es la dirección real de la persona que esta enviando los

---

[14] *Sendmail* es el programa más común de servidores de correo electrónico. Desde la versión 8.9.0 (19/5/1998) en su configuración por defecto rechaza los mensajes que no pertenecen al dominio de correo que administra.

[15] MAPS RBL (Realtime Blackhole List), ORDB (Open Relay DataBase), DUL (Dial-up User List), SPEWS (Spam Prevention Early Warning System), SBL (Spamhaus Block List), ORBZ (Open Relay Blackhole Zones), y muchas otras. Para más información ver Google - Spam Blacklists Index
http://directory.google.com/Top/Computers/Internet/Abuse/Spam/Blacklists/

[16] Para más información ver, EFFector Newsletter vol.14 num.31 “Electronic Frontier Foundation - Public Interest Position on Junk Email: Protect Innocent Users”
http://www.eff.org/effector/HTML/effect14.31.html#II .

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

mensajes[4], y trabaja de la siguiente manera. Cuando llega un mensaje proveniente de una dirección perteneciente a la *Lista Blanca*, éste es enviado a la bandeja de entrada del destinatario con normalidad. Si en cambio el mensaje proviene de una dirección desconocida, éste se guarda en una carpeta de mensajes "sospechosos", y se le envía al remitente un mensaje con un texto similar a este:

```
Ud. envió recientemente este mensaje de correo
electrónico:

     De: remitente@dominio.com
     Para: destinatario@otrodominio.com
     Fecha: Sábado 1 de Marzo de 2003 15:18:23 (HOA)
     Tema: Felices Vacaciones

El mismo no fue entregado a destino porque su dirección
de correo electrónico es desconocida para nosotros. Si
desea que el mismo sea entregado por favor responda a
este mensaje y ponga en el tema del mensaje "[CV:43123]".
Este es su código de verificación. Al recibir el mismo,
agregaremos su dirección de email a la lista de
direcciones habilitadas y entonces su correo será
entregado con normalidad.
```

Este esquema de *lista blanca* tiene la ventaja que se "autoadministra". Pero resulta algo engorroso para los que envían un mensaje por primera vez y tienen que completar el procedimiento de "validación" de su dirección de email. Además si el mensaje automático que envía el sistema por algún motivo no llega a destino (o no es visto por el destinatario), el mensaje legítimo que fue puesto en "cuarentena" se perderá. También es probable que se pierdan algunos mensajes legítimos que son enviados por programas en lugar de personas. Por ejemplo avisos de correo no entregado, respuestas automáticas de aviso de ausencia por vacaciones, etc.

Filtros por Contenido

La solución ideal al problema del Spam sería desarrollar un filtro que detecte y elimine este tipo de mensajes. Si bien esto no es posible de lograr con absoluta precisión, se han probado algoritmos que intentan acercarse a este objetivo.

Los primeros filtros de Spam se basaban en patrones que se comparaban con el cuerpo o encabezado del mensaje[17]. Estos patrones debían ser creados y mantenidos a mano y la eficacia de los mismos estaba ligada a la habilidad de la persona que los definía.

Más recientemente se comenzaron a estudiar técnicas de Aprendizaje Automático (*Machine Learning*), para el filtrado de este tipo de mensajes [SAH/98] [AND/00a] [CAR/01] [SAK/01], con resultados satisfactorios.

[17] *Procmail* es un programa que permite procesar email de acuerdo a si cumple o no con patrones definidos por el usuario. Este programa ha sido utilizado comúnmente para filtrar spam.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

2. Aprendizaje Automático y Clasificación de Documentos

La clasificación de documentos consiste en la asignación automática de etiquetas con temas a textos en lenguaje natural a partir de un conjunto predefinido de temas [SEB/02]. Actualmente se hace con distintas técnicas de Aprendizaje Automático. Esto es el empleo de procesos que aprenden en base a ejemplos de documentos previamente clasificados (normalmente llamado "conjunto de entrenamiento").

Las aplicaciones de la clasificación de documentos abarcan desde la asignación de palabras claves a documentos para su posterior recuperación, la organización de textos en un índice temático/jerárquico, o el filtrado de documentos.

Los métodos de clasificación de documentos constan de ciertas etapas, las que detallaremos a continuación.

2.1 Selección de Atributos

Los textos son normalmente representados como un vector de pesos de los términos (también llamados atributos) $\vec{d} = \langle w_1, w_2, \ldots, w_n \rangle$. Cada w_i es un número que representa la importancia del término *i* dentro del documento *d*, siendo *n* el número de términos utilizados para la clasificación.

Existen diferentes formas de interpretar que es un término y diferentes formas de otorgarles peso a los mismos.

Una forma usual es interpretar que los términos son palabras. Algunos experimentos en los que se intentaron representaciones más sofisticadas que ésta no dieron resultados convincentes [LEW/92].

Para representar los documentos rara vez se utilizan todas las palabras presentes. En cambio se selecciona un subconjunto de ellas. Esto se hace por dos razones. Por un lado algunos algoritmos (como ser LLSF, Redes Neuronales, etc.) aumentan el uso de CPU y memoria con la cantidad de términos. Y también muchas técnicas de clasificación pueden padecer el problema de *sobreajuste*. Esto es que el algoritmo aprende a identificar muy bien el conjunto de

entrenamiento, pero no funciona tan bien con ejemplos nuevos. Algunos estudios [FUH/91] parecen indicar que son necesarios alrededor de entre 50 y 100 documentos de entrenamiento por cada término utilizado.

Existen distintas técnicas desarrolladas para la reducción del espacio de atributos. Algunas de las más utilizadas se basan en aplicar a los atributos una función que mide su "importancia" para la tarea de clasificación, y luego seleccionar una cantidad fija entre los de mayor calificación. Idealmente los métodos de selección de atributos debieran descartar los términos irrelevantes y redundantes[18].

A veces antes de aplicar estas medidas de relevancia se eliminan los "conectores" (stop-words). Estas son palabras que no agregan significado al documento y suelen ser muy comunes (ej. artículos, preposiciones, pronombres, etc.). También se suele reemplazar las palabras por sus "raíces". La idea es agrupar las palabras que pertenecen a la misma familia de palabras bajo una denominación común. A esto se le llama "lematización" (stemming).

Algunas funciones de selección de atributos son[19]:

Frecuencia Documental

$$DF(t_k) = P(t_k)$$

Mide la cantidad de documentos en que aparece un término y da mayor puntaje a los términos más comunes. Esta medida de importancia se basa en la idea de que los términos poco comunes son también poco útiles a la hora de clasificar un documento. Sin embargo para aplicar esta técnica es necesario primero eliminar los "conectores". A veces esta técnica se usa en combinación con alguna otra. Por ejemplo, primero se eliminan las palabras que aparecen menos de *n* veces (ej. *n* igual a 5), y luego al resto se las evalúa con otra técnica de selección de atributos.

[18] Sin embargo la mayoría de los métodos hacen énfasis en eliminar términos irrelevantes, pero no así los redundantes. Si bien existe forma de detectar estos términos[KOL/96], estos algoritmos son más complejos y queda pendiente demostrar la utilidad práctica de los mismos frente a los algoritmos tradicionales.

[19] Presentaremos aquí las fórmulas de selección de atributos expresadas para nuestro caso particular de filtrado de Spam donde hay sólo dos clases de documentos (mensajes), Spam y Legítimos. Pero cabe aclarar que en el caso general (donde existen muchas categorías), la forma de usar estos métodos puede diferir un poco.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Ganancia de Información

$$IG(t_k) = \sum_{\substack{c \in \{Spam, Legitimo\} \\ t \in \{t_k, \overline{t_k}\}}} P(c,t) Log\left(\frac{P(c,t)}{P(c)P(t)}\right)$$

Esta fórmula (a veces también llamada *información mutua*), mide la distancia (o entropía relativa) entre la probabilidad conjunta (de la clase y el término) y el producto de las probabilidades[COV/91]. Indica qué tan dependientes son la clase y el término. Si la clase y el término son totalmente independientes el uno del otro, entonces se cumple que $P(c,t) = P(c)P(t)$, por lo que el resultado será cero.

Ji-cuadrado

$$\chi^2(t_k) = \frac{N\left(P(t_k, spam)P(\overline{t_k}, legit) - P(t_k, legit)P(\overline{t_k}, spam)\right)^2}{P(t_k)P(\overline{t_k})P(spam)P(legit)}$$

Siendo N el número total de documentos usados durante el entrenamiento.

Ji-cuadrado es una prueba que se suele utilizar para analizar si los resultados de un experimento difieren de una hipótesis previa. En este caso la hipótesis es que el término y la clase son independientes. Y esta hipótesis se compara contra los casos observados en el conjunto de entrenamiento [GAL/00].

Coeficiente GSS

$$GSS(t_k) = P(t_k, spam)P(\overline{t_k}, legit) - P(t_k, legit)P(\overline{t_k}, spam)$$

Este método de selección de atributos es una simplificación de χ^2 [GAL/00]. El mismo hace énfasis en los términos que tienen una correlación con la categoría (spam), y penaliza a los que están ligados al complemento de ella (legítimos). Esta fórmula fue probada frente al conjunto de datos `Reuters-21758` que consiste de 12902 documentos y 118 categorías. El mismo produjo buenos resultados cuando se lo utilizó con grandes niveles de reducción de términos.

Índice de Relevancia

$$RI(t_k) = \left(P(t_k, spam)P(\overline{t_k}, legit) - P(t_k, legit)P(\overline{t_k}, spam)\right)^2$$

En este trabajo definimos esta nueva fórmula para selección de atributos. La misma es también una simplificación de χ^2 (como el coeficiente GSS) pero conservamos el cuadrado de la fórmula original. La idea es que para la tarea de

filtrado, donde sólo existe una categoría (Spam), y donde la proporción de documentos que pertenecen a ella no es pequeña, nos interesan tanto los términos que tienen una asociación fuerte con el Spam, como aquellos que lo tienen con su complemento (los Legítimos).

2.2 Entrenamiento

En esta etapa se procesa un conjunto de documentos previamente clasificados, para que el algoritmo "aprenda" en forma inductiva las características de los mismos. Explicaremos aquí dos ejemplos de estos algoritmos

Naïve Bayes

Este clasificador se basa en un método probabilístico [SEB/02]. Calcula la probabilidad que un documento dado (representado por un vector de pesos de palabras) pertenezca a una categoría utilizando el teorema de Bayes.

$$P\left(spam\middle|\vec{d}_j\right)=\frac{P(spam)P\left(\vec{d}_j\middle|spam\right)}{P\left(\vec{d}_j\right)}$$

En esta fórmula $P(spam)$ se estima en base al conjunto de entrenamiento. Pero $P\left(\vec{d}_j\middle|spam\right)$ es problemático de calcular, debido a que el espacio de todos los documentos posibles es muy grande, y con los datos de entrenamiento no se tiene suficiente información para ello. Es aquí donde se hace la simplificación de pensar que cada palabra del documento es estadísticamente independiente de las otras[20]. En base a esto obtenemos que

$$P\left(\vec{d}_j\middle|spam\right)=\prod_{k=1}^{n}P\left(w_{kj}\middle|spam\right)$$

Calculándose $P\left(w_{kj}\middle|spam\right)$ en base a los datos de entrenamiento.

También es difícil de calcular $P\left(\vec{d}_j\right)$, pero este valor se puede despejar de la siguiente manera:

$$P\left(spam\middle|\vec{d}_j\right)+P\left(legit\middle|\vec{d}_j\right)=1$$

[20] Este supuesto no se cumple en la realidad. De allí el nombre de Naïve (inocente)

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

$$\frac{P(spam)P(\vec{d}_j|spam)}{P(\vec{d}_j)} + \frac{P(legit)P(\vec{d}_j|legit)}{P(\vec{d}_j)} = 1$$

$$P(spam)P(\vec{d}_j|spam) + P(legit)P(\vec{d}_j|legit) = P(\vec{d}_j)$$

Existen numerosos estudios de esta técnica para la clasificación de documentos [LEW/98] [MCC/98] [KIM/00] como así también algunos estudios en el filtrado de Spam [AND/00a] [AND/00b] [SAH/98].

Ajuste Lineal por Cuadrados Mínimos (LLSF)

Este método [YAN/94] se basa en encontrar una solución aproximada a un sistema de ecuaciones de la forma:

$$A \cdot x = b$$

Donde *A* es una matriz de *m* filas y *n* columnas. Siendo *m* el número de documentos del conjunto de entrenamiento, y *n* el número de términos que se consideraron relevantes en la etapa de selección de atributos. *b* es un vector columna de tamaño *m* donde cada fila indica si el mensaje correspondiente del conjunto de entrenamiento es o no Spam.

Rara vez este sistema tiene una solución exacta, por lo que se busca una solución aproximada. Este método lo veremos con más detalle en el capítulo 3.3.

2.3 Validación

Cuando se presenta un nuevo documento a ser clasificado la mayoría de estas técnicas retornan un valor que indica la factibilidad de que el documento pertenezca a la categoría en cuestión. Pero para poder concretamente clasificar el documento, necesitamos definir un "umbral" por encima del cual podemos decir que el documento pertenece a la categoría (es Spam).

Para algunos métodos de clasificación a veces es posible definir este umbral en forma analítica. Esto se puede hacer cuando existe algún modelo teórico que permita determinar cual es el valor óptimo para maximizar la medida de efectividad del algoritmo.

Cuando esto no es posible hay que recurrir a métodos experimentales que permitan hallar el valor del umbral. Para esto se utiliza un conjunto de datos

(distinto del conjunto de entrenamiento), llamado conjunto de validación, que es usado para ajustar éste valor. Tradicionalmente existen dos maneras de lograr dicho objetivo [SEB/02].

Un método consiste en aplicar el clasificador a los distintos documentos del conjunto de validación, y probar cada uno de los valores retornados por la función de clasificación hasta encontrar el que maximice la función de efectividad, las que serán vistas con mayor detalle más adelante.

Otro método consiste en aplicar la función de clasificación al conjunto de validación y seleccionar el umbral que divida estos documentos en dos clases con la misma proporción que la existente en el conjunto de entrenamiento [SEB/02].

2.4 Pruebas

La prueba del clasificador obtenido consiste en aplicar el mismo a un conjunto de datos de prueba (distinto al de entrenamiento y al de validación), y luego evaluar la bondad del mismo mediante alguna medida de efectividad.

Hay dos alternativas utilizadas normalmente para probar un clasificador. O dividir el conjunto de mensajes en dos partes, entrenamiento[21] y prueba, o dividir el conjunto en *k* partes y seleccionar uno de estos subconjuntos como conjunto de prueba y los restantes *k*-1 como conjuntos de entrenamiento. Luego se repite el experimento *k* veces alternando el conjunto de prueba seleccionado y se promedian los *k* resultados obtenidos. Esta última es llamada "validación cruzada en *k* partes".

Existen diversas medidas de efectividad habitualmente utilizadas. Algunas de ellas son:

Precisión

$$P = \frac{N_{legit \to legit}}{N_{legit \to legit} + N_{spam -> legit}}$$

Siendo $N_{A \to B}$, el número de mensajes que pertenecen a la clase *A* y el clasificador dijo eran de la clase *B*. La precisión se define como la proporción de

[21] Si la etapa de validación es necesaria, entonces hará falta subdividir el conjunto de entrenamiento, en un conjunto de entrenamiento propiamente dicho, y otro de validación.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

documentos clasificados correctamente sobre todos los que el clasificador dijo que pertenecían a la categoría.

Recuperación

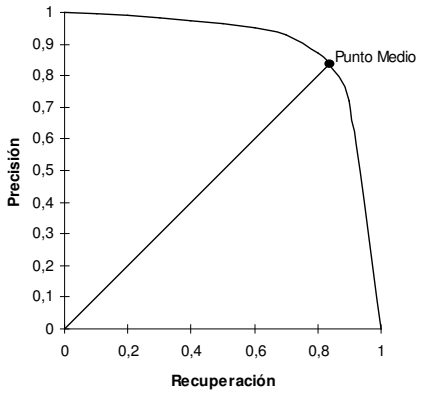
$$R = \frac{N_{legit \to legit}}{N_{legit \to legit} + N_{legit \to spam}}$$

La recuperación se define como la proporción de documentos clasificados correctamente sobre todos los que realmente pertenecían a la categoría.

Modificando el valor del umbral se puede mejorar la precisión a expensas de la recuperación y viceversa. Por lo que un valor aislado de precisión o recuperación no dice demasiado. Por eso se suelen definir otras medidas de efectividad que son una combinación de ambos valores.

Punto Medio

Como dijimos recién, es posible mejorar la precisión a expensas de la recuperación y viceversa. Se define "Punto Medio" al valor en que ambas medidas coinciden.



Función F_β

Para algunos problemas es más importante la recuperación que la precisión. Esto ocurre por ejemplo en el filtrado de spam, donde es mucho más importante no perder mensajes legítimos, que escape al filtro un mensaje spam. La función F_β trata de considerar esas situaciones.

$$F_\beta = \frac{\left(\beta^2 + 1\right) P \cdot R}{\beta^2 P + R}$$

En esta fórmula $\beta \geq 0$ representa la importancia relativa de la recuperación sobre la precisión. F_0 mide sólo la precisión, $F_{+\infty}$ mide sólo la recuperación y F_1 es una combinación que castiga de igual manera a ambos tipos de errores.

Exactitud

$$Acc = \frac{N_{legit \to legit} + N_{spam \to spam}}{N_{legit} + N_{spam}}$$

Es la proporción de mensajes bien clasificados sobre el total de mensajes. Para el caso de filtrado de spam, es mucho más importante que los mensajes legítimos sean bien clasificados, que los spam. Por ello se puede usar esta fórmula ponderada por un coeficiente λ (lambda).

$$WAcc = \frac{\lambda N_{legit \to legit} + N_{spam \to spam}}{\lambda N_{legit} + N_{spam}}$$

Error

$$Err = \frac{N_{legit \to spam} + N_{spam \to legit}}{N_{legit} + N_{spam}}$$

Es la proporción de mensajes mal clasificados sobre el total de mensajes. También se puede ponderar ambos tipos de errores mediante el coeficiente λ (lambda).

$$WErr = \frac{\lambda N_{legit \to spam} + N_{spam \to legit}}{\lambda N_{legit} + N_{spam}}$$

Razón de Costo Total

$$TCR = \frac{N_{spam}}{\lambda N_{legit \to spam} + N_{spam \to legit}}$$

Esta medida de efectividad [AND/00a] compara el error del clasificador contra el costo de no filtrar. Valor mayores a 1 indican que conviene usar esa técnica de filtrado, y menores que 1 que no es aconsejable usarlo.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

3. Presentación del problema

En el problema de clasificación de mensajes se presenta la necesidad de encontrar una transformación que dado un mensaje determine si el mismo pertenece o no a la categoría de spam. Dado un conjunto de mensajes de entrenamiento manualmente clasificados como spam o legítimos, debería ser posible aprender en forma empírica la relación existente entre las palabras que los componen con la categoría de spam.

El problema radica en que el conjunto de mensajes utilizados para entrenamiento es finito y nos encontramos con la realidad de que tendremos un conjunto infinito de mensajes para clasificar.

Buscamos una forma numérica de representar el conjunto de mensajes de entrenamiento y su correspondiente relación con la categoría de spam, para de esta forma, poder utilizar alguna técnica matemática que nos resuelva la transformación de un espacio en otro.

Representaremos al conjunto de mensajes utilizados para el entrenamiento como una matriz, que llamaremos A, donde cada fila es un mensaje, cada columna corresponde a una palabra, y cada posición de la matriz a_{ij} refleja el peso de la palabra j en el mensaje i. Por otro lado tendremos un vector columna, que llamaremos b, donde cada fila indica la relación entre la categoría de spam y el mensaje correspondiente a la misma fila de la matriz A.

3.1 Representación de los mensajes

Utilizamos distintos criterios al momento de armar la matriz A que representará a los mensajes. Como dijimos anteriormente, cada fila de la matriz representa a un mensaje distinto, mientras que cada columna se corresponde con una de las palabras que han sido consideradas relevantes por el algoritmo de selección de atributos. La idea es asignarle un peso en cada posición de la matriz según si la palabra correspondiente está o no presente en el mensaje.

Los criterios utilizados en el momento de asignar un valor a una posición de la matriz *A*, correspondiente a la aparición o no de una palabra en el mensaje, fueron:

- Valores binarios [YAN/94],

  $a_{ij} = 1$ si la palabra p_j esta presente en el iésimo mensaje y
  $a_{ij} = 0$ si no lo está.

- Frecuencia de la palabra en el mensaje (TF) [YAN/94][SEB/02],

  a_{ij} = número de veces que la palabra p_j aparece en el mensaje iésimo,

  cuanto más aparece una palabra en un mensaje, más representativa es de su contenido.

- Frecuencia inversa del documento (IDF) [YAN/94][SEB/02],

$$a_{ij} = \log\left(\frac{\textit{número total de mensajes}}{\textit{número de mensajes que contienen la palabra } p_j}\right) + 1$$

  una palabra es menos discriminante en cuanto más documentos aparece.

- Combinación de las dos opciones anteriores [YAN/94][SEB/02],

$$a_{ij} = \mathrm{TF}_{ij} \times \mathrm{IDF}_j$$

Adicionalmente, en este trabajo, proponemos los siguientes criterios de armado de la matriz de mensajes:

- Probabilidad de spam (PS)

  a_{ij} = Probabilidad de que el mensaje sea spam dado que tiene la palabra $P\left(Spam\middle|t_{ij}\right)$ si contiene la palabra.

  a_{ij} = Probabilidad de que el mensaje sea spam dado que no tiene la palabra $P\left(Spam\middle|\overline{t_{ij}}\right)$ si no la contiene.

- Probabilidad de spam y legítimo (PSL)

  a_{ij} = Probabilidad de que el mensaje sea spam dado que tiene la palabra $P\left(Spam\middle|t_{ij}\right)$ si contiene la palabra.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

a_{ij} = menos la probabilidad de que sea legítimo dado que no tiene la palabra $-P\left(Legit\middle|\overline{t_{ij}}\right)$ si no la contiene.

3.2 Representación de las categorías

A diferencia de muchos otros problemas de clasificación, en nuestro caso existe una única categoría que es la de spam. Por lo tanto debemos representar la pertenencia o no de un mensaje a dicha categoría. Para ello utilizamos un vector columna *b* en el cual cada posición indica la relación de pertenencia a la categoría de spam del mensaje representado por la misma fila en la matriz de mensajes *A*.

La cuestión es ver que valores utilizamos para indicar que un mensaje es spam o legítimo. Las pruebas realizadas fueron:

- b_i = 1 si el mensaje A_i es spam, -1 si es legítimo
- b_i = 1 si el mensaje A_i es spam, 0 si es legítimo

En ambos casos los resultados fueron muy similares, no observándose mejoras notables cuando utilizamos una u otra representación.

3.3 Resolviendo el problema utilizando LLSF

La matriz *A* junto con el vector *b* da la información de co-ocurrencia entre las palabras y la categoría de spam. Con estos elementos podemos calcular la transformación del espacio origen, el de mensajes, en el espacio destino, el de la categoría de spam. Esta transformación es una solución del problema LLSF, que en nuestro caso consiste en encontrar un vector x' que minimice la siguiente fórmula:

$$\left\| A \cdot x' - b \right\|_2$$

En teoría el problema LLSF tiene siempre al menos una solución. Un método para resolver dicho problema es utilizar la descomposición en valores singulares (SVD) [PRE/93].

Cada posición x'_j da un puntaje que relaciona la palabra en la columna *j* de la matriz *A* con la categoría de spam. Estos puntajes también son llamados "pesos", y a diferencia de los valores utilizados para armar la matriz *A* que indica-

ban una relación entre la palabra y el documento, estos relacionan una palabra con la categoría de spam.

3.3.1 Una aproximación al método SVD

Este método se basa en un teorema de álgebra lineal que dice que una matriz A de $m \times n$, donde el número de filas m es mayor o igual que el número de columnas n, puede ser escrita como el producto de una matriz U de $m \times n$ ortogonal por columnas, una matriz diagonal W de $n \times n$ con valores positivos o ceros, y una matriz traspuesta V de $n \times n$ ortogonal.

$$\begin{bmatrix} \\ A \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ U \\ \\ \\ \\ \end{bmatrix} \cdot \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & \dots & \\ & & & w_N \end{bmatrix} \cdot \begin{bmatrix} \\ V^T \\ \\ \\ \\ \end{bmatrix}$$

Las matrices U y V son ortogonales en el sentido de que sus columnas son ortonormales,

$$\sum_{i=1}^{M} U_{ik} U_{in} = \begin{cases} 0 \text{ si } k \neq n \\ 1 \text{ si } k = n \end{cases} \qquad \text{con } 1 \leq k \leq N,\ 1 \leq n \leq N$$

$$\sum_{j=1}^{M} V_{jk} V_{jn} = \begin{cases} 0 \text{ si } k \neq n \\ 1 \text{ si } k = n \end{cases} \qquad \text{con } 1 \leq k \leq N,\ 1 \leq n \leq N$$

Como V es cuadrada, luego también es ortonormal por filas.
Tenemos que,

$$U^T \,.\, U = V^T \,.\, V = V \,.\, V^T = I$$

Ahora basándonos en el planteo del método SVD , podemos decir que:

$$A = U \,.\, W \,.\, V^T$$

Dado que $U^T \cdot U = I$

$$U^T \,.\, A = W \,.\, V^T$$

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

dado que W es una matriz diagonal es trivial calcular su inversa

$$W^{-1} = [diag(1/w_j)] \quad ^{22}$$

por lo que

$$W^{-1}.U^T . A = V^T$$

también sabemos que $V \cdot V^T = I$

$$V.W^{-1}.U^T . A = \mathrm{I}$$

de esta forma tendríamos que

$$A^+ = V . W^{-1} . U^T$$

siendo A^+ la pseudoinversa de A. Para nuestro problema, conociendo A^+ fácilmente podríamos obtener el valor de x'

$$x' = A^+ \cdot b$$

en definitiva, conociendo el vector b y calculando V, W^{-1} y U^T utilizando el método SVD podemos obtener el valor de nuestro vector de pesos x'. Para esto se suele usar el algoritmo conocido como LINPACK [DON/79], el cual tiene una complejidad de $O(m\ n^2)$.

3.4 Aplicación del resultado

Una vez obtenido el vector x' que permite mapear los mensajes utilizados durante el entrenamiento en su correspondiente valor que lo identifica como spam o legítimo, podremos utilizarlo para evaluar los mensajes cuya pertenencia a la clase spam es para nosotros desconocida.

Cada nuevo mensaje será representado como un vector con las mismas características que posean las filas de la matriz A.

El valor obtenido de la multiplicación de dicho vector con nuestro vector x'

[22] Un problema que se puede presentar es si w_j=0, esto ocurre cuando las columnas de A no son linealmente independientes. En este caso se puede eliminar la columnas j de U y V, y la fila y columna j de W. O sino simplemente ignorar este w_j [PRE/93]

permitirá determinar si el mensaje que representa deberá ser tratado como spam o como un mensaje legítimo.

Significado del parámetro λ

Dado que en general es más importante que no nos equivoquemos en la clasificación de un mensaje legítimo contra la posibilidad de hacerlo con un spam, diremos que clasificar un mensaje legítimo como spam es λ veces más costoso [AND/00c][CAR/01]. Este valor servirá al momento de evaluar la efectividad del clasificador.

Alguno de los valores que podría tomar λ para reflejar la importancia que le demos a clasificar en forma errónea un mensaje legítimo con respecto a hacerlo con un spam podrían ser:

- $\lambda = 1$
  En este caso, el costo de clasificar mal cualquier mensaje, tanto spam como legítimo, es el mismo. Se utiliza en el caso que ningún mensaje sea descartado automáticamente y sólo es marcado indicando la condición del mismo.
- $\lambda = 9$
  Ahora aumentamos el costo de clasificar en forma errónea los mensajes legítimos. Por esta razón podríamos proceder a separar los mensajes que consideramos spam, pero permitiendo al destinatario verificar la condición de los mismos.
- $\lambda = 999$
  Es de suma importancia no clasificar como spam un mensaje que es legítimo, en este caso bloquear un mensaje legítimo es tan grave como que 999 mensajes de spam pasen el filtro. Por esta razón cuando utilizamos un valor de λ tan alto podemos proceder eliminando los mensajes que fueron clasificados como spam.

3.5 Algoritmo de clasificación

El algoritmo de clasificación puede ser dividido en varias etapas que colaboran en la obtención del vector x' y el valor del umbral.

Las pruebas fueron realizadas en el conjunto de datos Ling-Spam[23]

[23] El conjunto de datos Ling-Spam se encuentra disponible en http://www.aueb.gr/users/ion/lingspam_public.tar.gz

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

[AND/00b] compuesto de:

- 2412 mensajes obtenidos al azar de una lista de correo que trata temas de lingüística, y removiendo el texto agregado por el servidor de la lista.
- 481 mensajes de spam. Los archivos adjuntos, marcadores de HTML, y mensajes de spam duplicados recibidos en el mismo día no fueron incluidos.

Estos mensajes fueron divididos en 10 grupos, cada uno de ellos con aproximadamente 241 mensajes legítimos y 48 mensajes de spam.

Las etapas en las que se divide el algoriítmo son:

- 1ra Etapa – Selección de Atributos
  En esta etapa utilizaremos 6 grupos de mensajes.
  A partir de ellos seleccionaremos las palabras que permitirán decidir con mayor seguridad la clase a la cual pertenece un mensaje. Para ello utilizaremos alguno de los métodos de selección de atributos. Las palabras seleccionadas identificarán cada una de las columnas que componen la matriz A de mensajes.
- 2da Etapa – Entrenamiento
  En esta etapa se procesan los mismo grupos de mensajes que en la etapa anterior mediante el algoritmo LLSF para que el clasificador "aprenda" a distinguir los dos tipos de mensajes. El resultado de esta etapa es el vector x' que permite ponderar la importancia de cada palabra con la categoría de Spam.
- 3ra Etapa – Validación
  El objetivo de esta etapa es encontrar un valor de umbral que permita clasificar los mensajes desconocidos cometiendo el menor error posible. Se considerarán como spam aquellos mensajes que sean calificados con un valor mayor que el umbral obtenido y como legítimo al resto. En esta etapa utilizamos 3 grupos de mensajes (distintos a los anteriores), que serán clasificados utilizando el resultado obtenido en la etapa anterior. En función del resultado obtenido buscamos el valor de umbral que minimice el error en este conjunto de mensajes, teniendo en cuenta la importancia de mal clasificar un mensaje legítimo (parámetro λ).
- 4ta Etapa – Pruebas
  Esta es la etapa final que permite verificar los valores obtenidos por las etapas anteriores. El testeo se realiza sobre el último grupo de mensajes que no fue utilizado en ninguna de las etapas anteriores.

Cabe aclarar que para que los resultados no se vean influenciados por el subconjunto de mensajes utilizados, el algoritmo realiza 10 ciclos[24], utilizando en cada uno de ellos subconjuntos diferentes de mensajes para cada etapa. Los valores de testeo obtenidos en cada ciclo finalmente son promediados para así obtener un valor más veraz del funcionamiento del algoritmo.

3.6 Mejoras en el algoritmo

En función de mejorar los resultados en la clasificación de los mensajes hemos analizado distintos factores que podrían influir en la determinación de su condición de spam o de legítimo.

3.6.1 Funciones de selección de atributos

Un problema que se presenta al momento de tener que clasificar los mensajes es la cantidad de palabras diferentes que pueden componer a cada uno de ellos. Por esta razón, para poder determinar cuales son las palabras que mayor relevancia poseen, realizamos pruebas con diferentes funciones de selección de atributos. Estas funciones permitirán obtener aquellas palabras que mejor ayudarán a decidir si un mensaje, por presencia o ausencia de las mismas, podría ser considerado spam o legítimo.

Las pruebas realizadas utilizaron las siguientes funciones de selección de atributos:

- Ganancia de Información (IG)
- Ji-cuadrado (χ^2)
- Coeficiente GSS
- Índice de Relevancia

3.6.2 Sobreentrenamiento

Consiste en agregar una nueva etapa al algoritmo, anterior a la de pruebas. En esta etapa procedemos a realizar un nuevo entrenamiento pero esta vez con los 9 grupos de mensajes utilizados en las 2 etapas anteriores (entrenamiento y validación). De esta forma obtendremos como resultado un nuevo vector x' que será el utilizado en la etapa de pruebas.

[24] A esto se llama "Validación Cruzada en 10 partes". Ver sección 2.4

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Para esto consideramos dos posibles alternativas:

Utilizando las mismas palabras

Esta etapa consiste en realizar un nuevo entrenamiento pero esta vez utilizando los mensajes que fueron empleados durante las etapas de entrenamiento y validación. De esta forma contaremos con mayor información para la obtención del vector x' utilizado en la etapa de pruebas. Cabe aclarar que se utilizarán las mismas palabras que se consideraron relevantes en la etapa de selección de atributos.

Utilizando palabras nuevas

Ahora el procedimiento es similar al caso anterior, la diferencia está en que previamente se realiza una nueva selección de palabras sobre los conjuntos de mensajes utilizados durante el entrenamiento y validación, y se sobreentrena utilizando los nueve conjuntos de mensajes con esta nueva selección de palabras relevantes.

3.6.3 Armado de la matriz de mensajes para el caso binario

Una de las formas de armado de la matriz de mensajes en el caso de utilizar valores binarios es utilizar las reglas antes mencionadas, es decir, colocar un 1 o un 0 si el mensaje correspondiente tiene o no la palabra. En este trabajo definimos y evaluamos otras posibilidades para la asignación de los valores teniendo en cuenta si nos brinda más información que el mensaje contenga o no la palabra.

Considerando la probabilidad de spam

Supongamos que un mensaje que contenga la palabra tiene una probabilidad de 0,2 de ser spam, pero si no la contiene la probabilidad es de 0,9. En este caso podemos pensar que brinda más información el hecho de no tener la palabra que el de tenerla, diríamos con mayor seguridad que si un mensaje no contiene la palabra es probable que sea spam.

De esta forma una modificación alternativa a la construcción de la matriz de mensajes en formato binario es basarnos en este cálculo de probabilidades

$$Si\ P\left(Spam\middle|t_j\right) \geq P\left(Spam\middle|\overline{t_j}\right) \rightarrow \begin{cases} 1 & si\ el\ mensaje\ contiene\ la\ palabra \\ 0 & sino \end{cases}$$

$$Si\ P\left(Spam\middle|t_j\right) < P\left(Spam\middle|\overline{t_j}\right) \rightarrow \begin{cases} 0 & si\ el\ mensaje\ contiene\ la\ palabra \\ 1 & sino \end{cases}$$

Siendo $P\left(Spam\middle|t_j\right)$ la probabilidad de que el mensaje sea Spam dado que tiene la palabra *j*, y $P\left(Spam\middle|\overline{t_j}\right)$ la probabilidad de que lo sea dado que no tiene la palabra.

Considerando la probabilidad más alejada de 0,5

Por otro lado es posible evaluar qué brinda mayor información, si la presencia o la ausencia de una palabra en un mensaje. Si $P\left(Spam\middle|t_j\right)$ es un valor lejano a 0,5 entonces decimos que nos aporta más información que si es un valor cercano a 0,5. Por ejemplo, si sabemos que un mensaje que contiene la palabra t_j tiene una probabilidad de ser spam de 0,5, y por otro lado tenemos que un mensaje que no contiene la palabra t_j tiene una probabilidad de ser spam de 0,1 (lo cual implica que tiene una probabilidad de ser legítimo de 0,9). De estos valores podemos decir con mayor seguridad que si el mensaje no posee la palabra es muy probable que sea legítimo, mientras que si la tiene la probabilidad está igualmente repartida entre spam y legítimos. Tomando esta medida de importancia podemos decir que:

$$Si\ \left|P\left(Spam\middle|t_j\right)-0{,}5\right| \geq \left|P\left(Spam\middle|\overline{t_j}\right)-0{,}5\right| = \begin{cases} 1\ si\ el\ mensaje\ contiene\ la\ palabra \\ 0\ sino \end{cases}$$

$$Si\ \left|P\left(Spam\middle|t_j\right)-0{,}5\right| < \left|P\left(Spam\middle|\overline{t_j}\right)-0{,}5\right| = \begin{cases} 0\ si\ el\ mensaje\ contiene\ la\ palabra \\ 1\ sino \end{cases}$$

Considerando la probabilidad de la palabra

Podemos pensar que si la matriz de mensajes contiene muchos 1s y pocos 0s, el algoritmo utilizará una mayor cantidad de coeficientes en la determinación de la condición del mismo. Por lo que ninguna palabra en forma aislada tendrá un efecto determinante en la evaluación del mensaje, por el contrario, el resultado obtenido será una combinación de un gran número de términos.

$$Si\ P\left(t_j\right) \geq 0{,}5 = \begin{cases} 1\ si\ el\ mensaje\ contiene\ la\ palabra \\ 0\ sino \end{cases}$$

$$Si\ P\left(t_j\right) < 0{,}5 = \begin{cases} 0\ si\ el\ mensaje\ contiene\ la\ palabra \\ 1\ sino \end{cases}$$

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

3.6.4 Diferenciación entre el tema y el cuerpo del mensaje

Nos preguntamos si habría que diferenciar las palabras que aparecen en el tema de las que lo hacen en el cuerpo del mensaje. A modo de ejemplo consideremos la palabra "gratis", ¿tendrá tanto peso si esta palabra aparece en el cuerpo del mensaje como si lo hace en el encabezado? Sin realizar un análisis profundo pensamos que una palabra "sospechosa", como podría ser "gratis", apareciendo en el tema del mensaje podría ser más determinante de su condición de spam que si lo hiciera en el cuerpo. Esto se debe a que hay muchas palabras que buscan despertar la curiosidad del destinatario del mensaje para que éste sea leído, y por lo tanto tienen una alta frecuencia de aparición en el tema. Por otra parte, normalmente la cantidad de palabras que componen el tema es mucho menor al tamaño del mensaje propiamente dicho, por lo tanto, su aparición en el tema toma un valor de relevancia diferente que si lo hace el cuerpo del mensaje.

Pero más allá de toda especulación realizamos las pruebas necesarias para comprobar como se comportaba el algoritmo cuando se hacia esta diferenciación y cuando esto no ocurría.

3.6.5 Determinación del umbral

Debíamos encontrar el valor del umbral que mejor nos permitiera clasificar los mensajes, considerando que aquellos que obtuvieran un puntaje mayor que dicho valor serían considerados spam, mientras que el resto pasarían como legítimos.

Los métodos utilizados para determinar ese valor fueron:

Umbral fijo

En este caso, tomamos como umbral el valor medio entre los valores que definen a un mensaje como spam o como legítimo. En nuestro caso los mensajes que son spam tienen un valor de 1 mientras que los legítimos tienen un valor de –1. Por lo tanto nuestro umbral será el 0. Los mensajes que tomen valores mayores a 0 serán considerados spam, los menores a dicho número serán legítimos. Debido a que el valor del umbral es fijo no es necesaria una etapa de validación.

Buscando el punto óptimo

Ahora, utilizando la matriz que representa a los mensajes que destinamos al ajuste del umbral, y multiplicándola por el vector x' obtenido durante el entre-

namiento, obtendremos un vector resultado, llamémoslo *r.* Este vector nos servirá para buscar el valor del umbral con el cual obtendremos el menor error de clasificación.

$$\begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{array} \overset{\begin{array}{cccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{array}}{\overset{A}{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}}} \cdot \overset{x'}{\begin{bmatrix} 0{,}2 \\ -0{,}1 \\ 1 \\ -2 \\ 1{,}5 \\ 0{,}3 \end{bmatrix}} = \overset{r}{\begin{bmatrix} 1{,}7 \\ 0{,}1 \\ 2 \\ -2{,}1 \end{bmatrix}} \quad \overset{b}{\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}}$$

A = matriz que representa los mensajes que utilizaremos para realizar el ajuste del umbral (representación binaria)

x' = vector obtenido en la etapa de entrenamiento

r = vector resultado de la etapa de ajuste

b_i = 1 si el mensaje m_i es spam, -1 si es legítimo

La idea es ordenar el vector *r* y el vector *b* (el cual indica realmente de que tipo es cada uno de los mensajes que componen la matriz) en forma ascendente, utilizando los valores de *r* como referencia para el ordenamiento.

En nuestro ejemplo los vectores *r* y *b* luego del ordenamiento quedarían de la siguiente manera:

$$r_{\text{ordenado}} = [-2.1, 0.1, 1.7, 2.0]$$

$$b_{\text{ordenado}} = [-1.0, -1.0, 1.0, 1.0]$$

Inicialmente asignaremos al umbral un valor superior a cualquiera de los que podría ocurrir en el vector *r,* llamemos a este valor *U.* Consideraremos a todos los mensajes cuyo resultado, obtenido del vector $r_{ordenado}$, nos dio mayor o igual que *U* como spam y a los de menor resultado como legítimos. De esta forma partimos de la consideración de que todos los mensajes son legítimos (puesto que a todos les fue asignado un valor menor). Nuestro error inicial pasará a ser igual a la cantidad de mensajes de spam utilizados en la etapa de ajuste.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Ahora iremos recorriendo secuencialmente de derecha a izquierda el vector $r_{ordenado}$, y consideraremos como nuevo valor de U al hallado en dicho vector. De esta forma cada vez que modificamos nuestro U existen 2 posibilidades, que el mensaje relacionado con la posición de $r_{ordenado}$ fuera spam o que fuera legítimo. En el primer caso el error disminuiría en una unidad (dado que al decrementar el valor del umbral estamos clasificando correctamente un mensaje que es Spam). En cambio si el mensaje relacionado con la posición del vector $r_{ordenado}$ se corresponde a un mensaje legítimo el error se incrementa en λ (dado que ahora pasamos a clasificar como Spam un mensaje que no lo es).

El proceso continuaría decrementando el error cada vez que encontramos un mensaje que es spam, e incrementándolo en λ cada vez que encontramos un mensaje legítimo. De esta forma seleccionaríamos un valor de umbral igual al de la posición del vector $r_{ordenado}$ en la que el error fuera menor. El mejor valor hallado con este método determinaría el mejor umbral para los mensajes utilizado en la etapa de ajuste, y en definitiva será el utilizado en el momento de testear si un mensaje es spam o no.

Paso	**Valores del umbral *U***	**Valor del error en cada paso**
Inicio	$U > 2$	2 (todos los spam son mal clasificados)
1	$U = 2$	1 (un spam es considerado legítimo)
2	$U = 1{,}7$	0 (todos los mensajes son bien clasificados)
3	$U = 0{,}1$	λ (un mensaje legítimo es mal clasificado)
4	$U = -2{,}1$	2λ (dos mensajes legítimos son mal clasif.)
Final	$U = 1{,}7$	0 error final

En nuestro ejemplo U tomará los valores de 2, 1.7, 0.1 y –2.1. Inicialmente nuestro error es de 2 ya que consideramos como legítimos los 2 primeros mensajes que en realidad son spam. Cuando U toma valor 2 nuestro error disminuirá dado que el primer mensaje es spam, y todos los mensajes con valor mayor o igual serán considerados spam. Cuando U toma valor 1.7, nuevamente el error disminuye ya que el segundo mensaje también es spam. Ahora cuando U toma el valor de la segunda posición del vector r_{ordenado}, 0.1, el error se incrementa ya que tomaremos como spam a todos los mensajes que tengan valor superior o igual a 0.1, y en este caso estaríamos evaluando incorrectamente a este mensaje, ya que el mismo es legítimo. Finalmente cuando U toma valor –2.1, el error

se incrementa nuevamente debido a la evaluación incorrecta del último mensaje. En definitiva el umbral quedaría fijado en el valor de 1.7, considerando como spam a todos aquellos mensajes que obtengan un valor igual o superior al umbral hallado.

El algoritmo sería el siguiente:

```
umbral = +∞
Error = cantSpam // Cantidad ponderada de mensajes mal
                 // clasificados
menorError = Error
for j=1 to tamaño de r do
   begin
     if bOrdenado[j] = SPAM then
     begin
       Error = Error - 1
       if Error < menorError then
       begin
          menorError = Error
          umbral = r[j]
       end
    end
    else Error = Error + λ;
   end
```

Buscando el punto óptimo y ajustando por el valor de λ

En este caso aplicaremos el mismo método que en el caso anterior pero una vez obtenido el valor del umbral, lo ajustaremos en función del valor de λ.

El valor del umbral obtenido mediante el algoritmo anterior siempre coincide con uno de los valores asociados a los mensajes del conjunto de validación, llamémoslo r_i [25]. Ahora sería posible que el umbral óptimo fuese cualquier valor entre r_i y r_{i+1} . Por lo que proponemos ajustar el valor obtenido mediante el algoritmo anterior en base al coeficiente λ.

$$umbral = r_i - \left(\frac{r_i - r_{i+1}}{1 + \lambda} \right)$$

El objetivo sería buscar un valor intermedio entre el r_i que nos daba el menor error y r_{i+1}, teniendo en cuenta la importancia que le damos a mal clasificar mensajes legítimos contra la posibilidad de hacerlo con un spam. En el caso de

[25] Para facilitar la notación llamaremos r al vector $r_{ordenado}$

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

$\lambda = 999$, el valor del umbral se modificará poco dado que disminuyendo el valor del mismo corremos el riesgo de que más mensajes legítimos sean mal clasificados. Y cuando $\lambda = 1$, el umbral tomará el valor medio entre r_i y r_{i+1} ya que en este caso consideramos que mal clasificar un spam es tan importante como hacerlo con un mensaje legítimo.

Cálculo matemático del umbral.

Arampatzis et al. [ARA/00] proponen un método para calcular el umbral de un filtro basado en la suposición que el puntaje asignado a los documentos relevantes se puede modelar con una distribución normal y de los irrelevantes con una exponencial.

Sin embargo en nuestras pruebas no observamos dicho comportamiento. Notamos que si interpretamos el valor calculado por la función de clasificación como una variable aleatoria, la distribución de la misma se aproxima bastante a una campana de Gauss tanto para los Spam como los Legítimos[26].

Haciendo uso de esta propiedad divisamos una nueva forma de determinar el valor óptimo para el umbral. El método consiste en aplicar el clasificador que se obtuvo en la etapa de entrenamiento a los distintos mensajes del conjunto de validación, y en base a estos resultados se calcula el umbral de la siguiente manera:

$$t = \frac{\mu_L \sigma_S^2 - \mu_S \sigma_L^2 + \sigma_L \sigma_S \sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2) Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}}{\sigma_S^2 - \sigma_L^2}$$

[26] Esta discordancia con respecto a los resultados obtenidos en [ARA/00] creemos que se debe a las diferencias del método. Arampatzis et al. utilizan una técnica de recuperación de información basada en el algoritmo de Rocchio, mientras que nosotros utilizamos un algoritmo de clasificación basado en LLSF.

Siendo:

μ_L Valor medio de los mensajes Legítimos

μ_S Valor medio de los mensajes Spam

σ_L Desvío estándar de los mensajes Legítimos

σ_S Desvío estándar de los mensajes Spam

Para ver una explicación más detallada de cómo se llego a obtener esta fórmula remitirse al Apéndice I.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

4. Resultados obtenidos

A continuación mostraremos los resultados obtenidos por el algoritmo LLSF. En todos los casos, salvo aclaración, se cumplirán las siguientes condiciones:

- El vector *b* inicial toma valores -1 para identificar a los mensajes legítimos y 1 para los spam.
- Diferenciamos las palabras del cuerpo del mensaje con las que se incluyen en el tema.
- Utilizamos Índice de Relevancia como función de selección de atributos.
- Usamos el método matemático para calcular el umbral.
- Sobreentrenamos seleccionando palabras nuevas.
- Evaluamos si es más importante tener el término o no tenerlo al momento de construir la matriz de mensajes en formato binario.
- En el caso de utilizar Ganancia de Información para la selección de palabras, primero eliminamos aquellas que aparecen menos de 6 veces en todos los mensajes [SEB/02].

Para el caso en que λ toma valores 1 ó 9, obtuvimos mejores resultados utilizando una selección de 300 palabras, mientras que cuando su valor fue 999, los mejores resultados los obtuvimos con 400 palabras. Por esta razón en los siguientes gráficos utilizaremos dichos parámetros para mostrar los resultados obtenidos con las mejoras del algoritmo.

Como medida comparativa utilizaremos TCR, que expresa la mejora obtenida con la utilización del algoritmo LLSF contra no aplicar ningún filtro, permitiendo que todos los mensajes lleguen a su destinatario sin ser evaluados. Recordamos que el TCR muestra la relación existente entre los errores ponderados del caso base, sin aplicar ningún filtro, y el cometido por el algoritmo. Sabiendo que,

$$WErr^{b} = Error\ Ponderado\ en\ el\ Caso\ Base = \frac{N_S}{\lambda \cdot N_L + N_S}$$

$$WErr = Error\ Ponderado = \frac{\lambda \cdot N_{L \to S} + N_{S \to L}}{\lambda \cdot N_L + N_S}$$

se define

$$TCR = \frac{WErr^b}{WErr} = \frac{N_S}{\lambda \cdot N_{L \to S} + N_{S \to L}}$$

Valores mayores del TCR indican una mejor performance con respecto al caso base. Un valor de TCR < 1 indica que es mejor no aplicar el filtro.

4.1 Composición de la matriz *A*

De acuerdo a lo visto en la sección 3.1, los resultados obtenidos en función de los valores asignados a la matriz de mensajes fueron

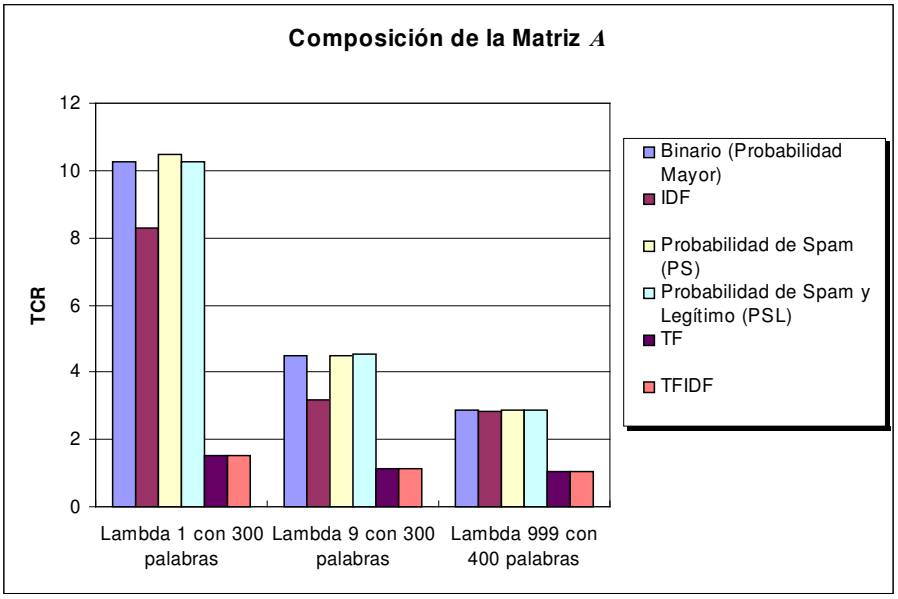


Fig. 1 – Diferentes formas de armar la matriz *A*

Pueden observarse resultados muy parejos para el caso de asignarle valores binarios o probabilísticos a la matriz de mensajes, por lo que de ahora en más mostraremos como influyen las mejoras para el caso de la matriz de mensajes compuesta por valores binarios.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

4.2 Funciones de selección de atributos

Resultados obtenidos luego de aplicar diferentes métodos de selección de atributos.

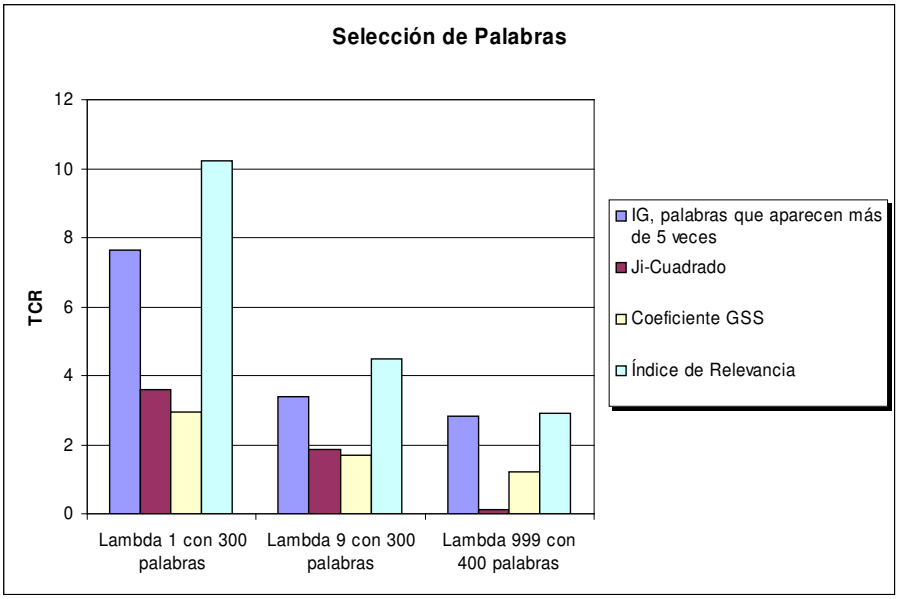


Fig. 2 – Funciones de Selección de Palabras

Se destaca claramente sobre los demás métodos la selección de atributos utilizando índice de relevancia. Recordamos que dicho método fue propuesto en este trabajo, y tiene como fórmula

$$RI(t_k) = \left(P(t_k, spam) P(\overline{t_k}, legit) - P(t_k, legit) P(\overline{t_k}, spam) \right)^2$$

para más detalles ver sección 2.1.

4.3 Sobreentrenamiento

Resultados obtenidos en el caso de sobreentrenar utilizando siempre las mismas palabras, realizando una nueva selección de palabras y sin sobreentrenar.

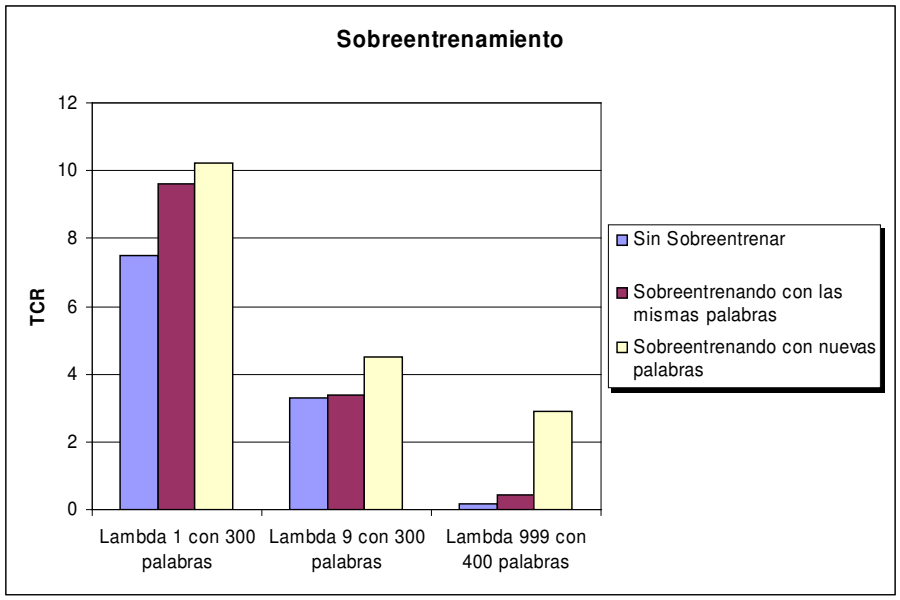


Fig. 3 – Resultados del sobreentrenamiento

Debido a que el sobreentrenamiento brinda más información al algoritmo, dado que utiliza mayor cantidad de mensajes, es que puede notarse la mejora obtenida con la inclusión de esta nueva etapa. En el gráfico también se destaca una mejora aún mayor cuando se seleccionan nuevas palabras sobre todos los mensajes que son utilizados en esta etapa. Para más detalles ver sección 3.6.2.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

4.4 Armado de la matriz de mensajes para el caso binario

Como explicamos anteriormente asignarle a la matriz de mensajes 1 ó 0 en función de si un mensaje tuviera o no una de las palabras seleccionadas es sólo una de las formas de construcción de la matriz de mensajes con valores binarios. También tuvimos en cuenta la posibilidad de que la presencia o ausencia de una palabra nos diera mayor información sobre la condición de spam de un mensaje. En la figura 4 mostramos los resultados obtenidos.

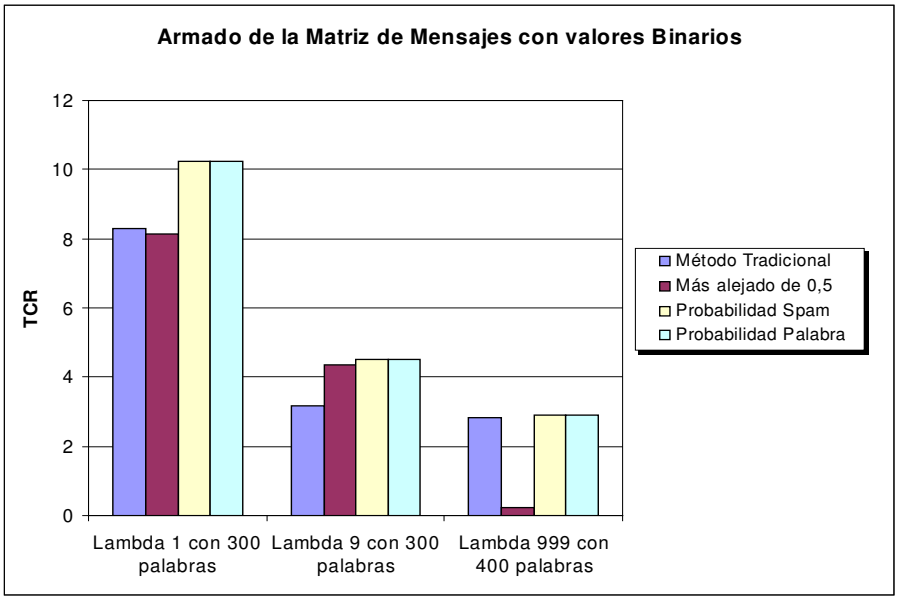


Fig. 4 – Diferentes formas de asignarle valores a la matriz de mensajes en el caso binario.

Puede observarse que los mejores resultados se obtuvieron con “Probabilidad de Spam” y “Probabilidad de la Palabra”. Los resultados para ambos métodos fueron idénticos. Luego de analizar esto notamos que “Probabilidad de Spam” también tiene la peculiaridad de asignar muchos 1s en la matriz de mensajes. Para más detalles ver sección 3.6.3.

4.5 Diferenciando el cuerpo y el tema del mensaje

Resultados obtenidos cuando diferenciamos las palabras que aparecen en el cuerpo del mensaje de las que lo hacen en el tema.

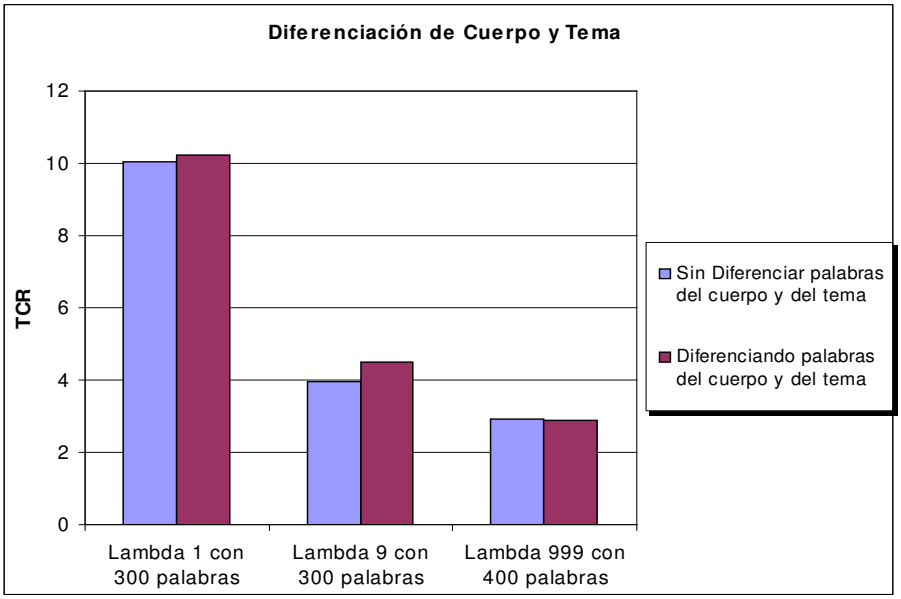


Fig. 5 – Diferenciando el cuerpo y el tema del mensaje

En el caso de λ igual a 1 ó 9, pudimos observar una pequeña mejora diferenciando las palabras que componen el cuerpo del mensaje de las que aparecen en el tema, no ocurriendo lo mismo para λ valiendo 999, donde los resultados fueron muy similares, aunque inferiores. Para detalles del porqué consideramos estas diferencias ver sección 3.6.4.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

4.6 Determinación del umbral

Resultados obtenidos utilizando diferentes métodos para el cálculo del umbral, el que determinará como spam a todos los mensajes que superen dicho valor.

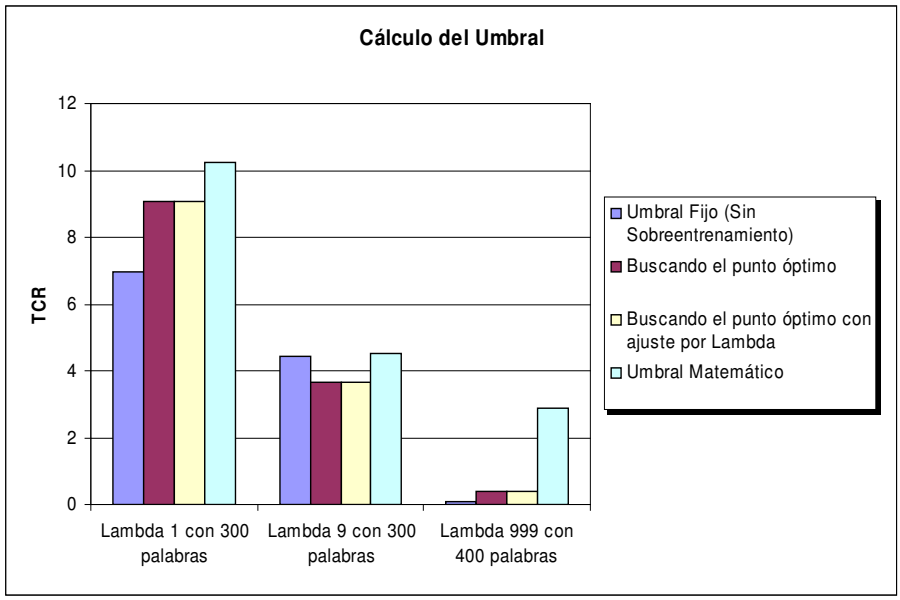


Fig. 6 – Cálculo del umbral

En este caso el método matemático, propuesto en este trabajo, que modela la distribución del puntaje asignado a los mensajes de spam y legítimos con una distribución normal, es el que permitió obtener mejores resultados. Para más detalles sobre las diferentes formas de cálculo del umbral ver sección 3.6.5.

4.7 Variando la cantidad de palabras

Resultados observados a partir de la variación en la cantidad de palabras seleccionadas para representar los mensajes.

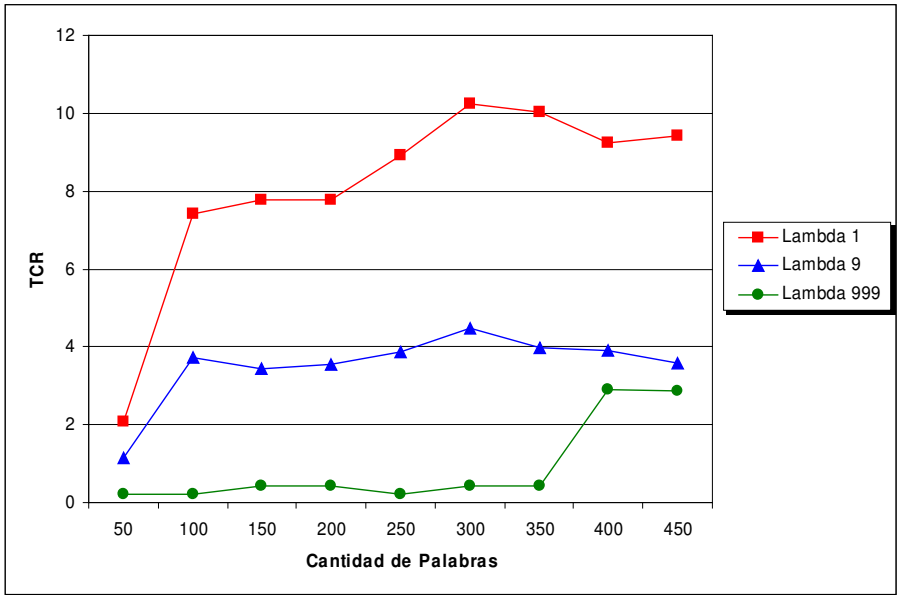


Fig. 7 – TCR obtenido variando el número de palabras seleccionadas

Los mejores resultados en el caso de λ igual a 1 ó 9 los obtuvimos con una selección de 300 palabras, mientras que para λ igual 999 necesitamos 400.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

4.8 Comparativa

Realizaremos un cuadro comparativo de los resultados obtenidos mediante LLSF y los presentados por Ion Androutsopoulos et al. en un estudio realizado sobre Aprendizaje Automático [AND/00b]. Este trabajo se basó en la implementación de un filtro Naïve-Bayesiano, utilizando Ganancia de Información para la selección de atributos. Las pruebas en este estudio fueron realizadas sobre el mismo conjunto de mensajes Ling-Spam.

Los resultados fueron los siguientes:

	Naïve-Bayes			**LLSF**		
λ	**Recuperación de Spam**	**Precisión de Spam**	**TCR**	**Recuperación de Spam**	**Precisión de Spam**	**TCR**
1	81,10%	96,85%	4,63	93,77%	96,37%	10,234
9	76,94%	99,46%	3,73	85,24%	99,03%	4,495
999	73,82%	99,43%	0,23	65,49%	100,00%	2,898

Tabla 1. Valores comparativos de los dos métodos

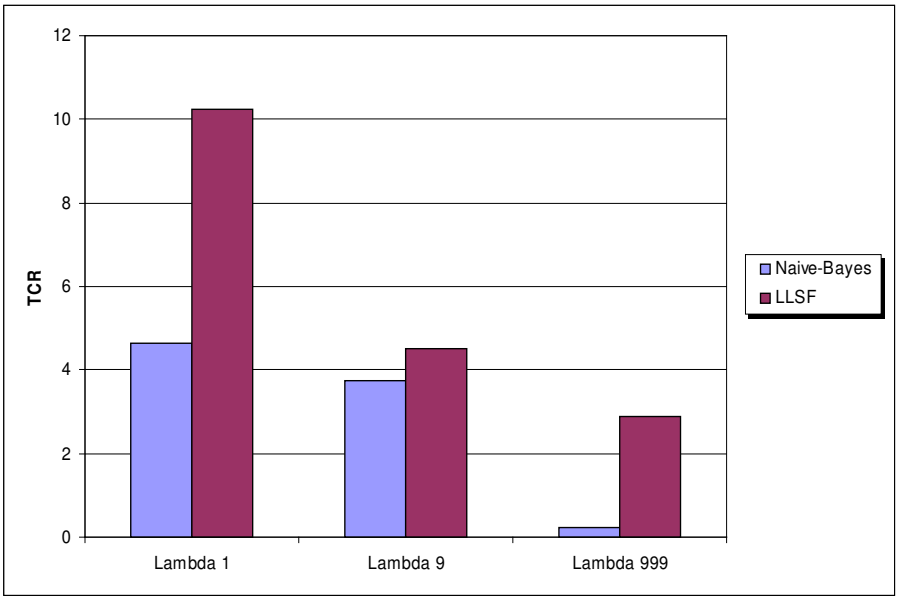


Fig. 8 - TCR obtenido con cada método

Para el caso de λ tomando valor 1, es decir cuando es lo mismo mal clasifi-

car un spam o un legítimo, podemos observar un mayor porcentaje de filtrado de spam aplicando LLSF, incrementando en más de un 12% los mensajes spam correctamente evaluados. Por otro lado hay una pequeña disminución en la precisión, en menos de un 0.5 %, indicando una mayor cantidad de mensajes legítimos considerados spam. Cabe aclarar que para este valor de λ evaluar en forma errónea un mensaje legítimo no es más grave que hacerlo con un spam. La mejora de LLSF puede observarse en el valor alcanzado por el TCR que es más del doble del valor obtenido utilizando Naïve-Bayes.

Cuando λ toma valor 9, es decir cuando es 9 veces más costoso evaluar incorrectamente un mensaje legítimo que hacerlo con un spam, nuevamente el valor de recuperación es notablemente mayor, en más de un 8%, indicando que mayor cantidad de mensajes spam no llegaran al destinatario. El porcentaje correspondiente a la precisión disminuye levemente, menos de un 0.5%. Finalmente podemos observar una pequeña mejora utilizando LLSF al comparar los valores de TCR obtenidos.

Con λ igual a 999, cuando es fundamental no evaluar incorrectamente los mensajes legítimos, pudimos obtener una precisión del 100%, es decir que todos los mensajes que fueron considerados spams realmente lo eran. Esto permitió alcanzar un valor de TCR muy superior al obtenido utilizando Naïve-Bayes, el cual no alcanzó la precisión óptima.

Más allá de todos los valores mostrados en la tabla, es fundamental remarcar que es el TCR el que debe ser utilizado para comparar los 2 métodos, ya que éste toma en cuenta el valor de λ para su cálculo, considerando la importancia que se le da a los 2 tipos de errores que el algoritmo puede cometer, filtrar mensajes legítimos o dejar pasar mensajes que son spam.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

5. Conclusiones y Trabajos Futuros

En este trabajo de tesis hemos analizado la factibilidad de utilización del algoritmo de clasificación LLSF [YAN/94] para el filtrado de spam. Hemos propuesto una serie de variantes al mismo que resultaron en una mejora en la perfomance.

- La función de selección de atributos "Índice de Relevancia", propuesta en este trabajo, produjo una mejora sustancial de la eficacia del filtro frente a otras funciones utilizadas con anterioridad (ver sección 4.2).
- Hemos introducido una nueva etapa en el proceso de aprendizaje del algoritmo, al que llamamos "sobreentrenamiento", que permitió obtener mejores resultados (ver sección 4.3).
- Presentamos nuevas alternativas de armado de la matriz de mensajes, "PS" y "PSL", que también resultaron superiores a otras técnicas antes estudiadas (ver sección 4.1)
- Para el caso del armado de la matriz con valores binarios, propusimos variantes alternativas al método tradicional dos de las cuales significaron una mejora en los resultados (ver sección 4.4)
- Para la determinación del umbral formulamos un método matemático basado en un análisis probabilístico de los valores retornados por el clasificador que permitió obtener un incremento en la eficacia del filtro (ver sección 4.6).

En base a estos resultados concluimos que es viable la utilización del algoritmo LLSF para el filtrado de spam, obteniéndose mejoras frente a Naïve-Bayes el cual ha sido estudiado y es usado actualmente con este fin.

Queda pendiente para un trabajo futuro el estudio de otras alternativas que pudiesen ayudar en la tarea de filtrado, como ser:

- La utilización de atributos no textuales, tales como, el dominio del remitente, la fecha hora del mensaje, si contiene marcadores HTML, si trae documento adjuntos, idioma, programa usado para enviar el mensaje, etc.

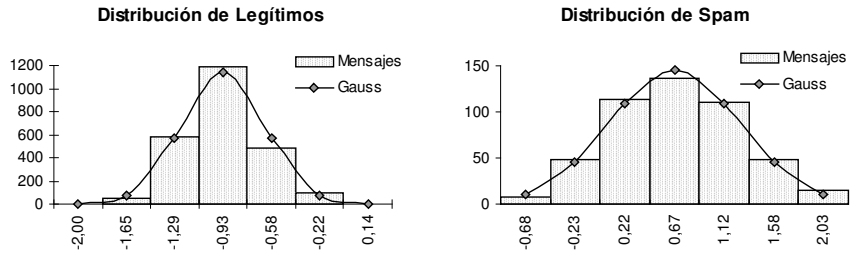
- La extracción automática de atributos no textuales. Los atributos mencionados en el punto anterior deben ser definidos manualmente, cabe preguntarse si es posible identificar alguno de ellos en forma automática.
- Con respecto a la diferenciación de palabras provenientes del tema y cuerpo del mensaje, no pudimos llegar a un resultado concluyente. En algunos casos obtuvimos leves mejoras y en otros no. Sería conveniente repetir estas experiencias con otros juegos datos.
- Sería interesante poder determinar si algunas de las mejoras aquí presentadas son también de utilidad en un espectro más amplio de problemas de categorización, más allá del filtrado de mensajes de correo electrónico.
- Cabe preguntarse si la utilización de la función de selección de atributos presentada en este trabajo "Índice de Relevancia", también resultaría en una mejora al algoritmo Naïve-Bayesiano, y de ser así, si estos nuevos resultados superarían a los aquí obtenidos mediante LLSF.

Como complemento a estas técnicas podrían utilizarse elementos adicionales que brindan una mayor flexibilidad a estos métodos, como ser la posibilidad de reconocer ciertas direcciones de email que serán evaluadas como legitimas o spam (listas blancas y negras), y la habilidad de considerar en la evaluación de un mensaje la historia de ese remitente, ya que si alguien no suele enviar mensajes spam, es poco probable que de repente comience a hacerlo.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Apéndice I

En nuestras pruebas notamos que si interpretamos el valor calculado por la función de clasificación como una variable aleatoria, la distribución de la misma se aproxima bastante a una campana de Gauss tanto para los Spam como para los Legítimos.



Haciendo uso de esta propiedad divisamos una nueva forma de calcular el valor óptimo para el umbral del clasificador.

Suponiendo que tanto los mensajes spam como los legítimos tienen una distribución normal

$$f(x) = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}} \qquad \sigma > 0 \quad \text{Desvío estándar} \qquad \mu \quad \text{Valor medio}$$

Y si llamamos *t* al umbral utilizado, resulta que la probabilidad que un mensaje legítimo sea mal clasificado como spam es:

$$P_{L\to S}(t) = \int_t^{\infty} f_L(x)dx$$

Y la probabilidad que un spam sea mal clasificado como legítimo

$$P_{S\to L}(t) = \int_{-\infty}^{t} f_S(x)dx$$

Si N_L y N_S son la cantidad de mensajes legítimos y spam respectivamente en nuestro conjunto de ajuste, el número esperado de mensajes legítimos mal clasificados es

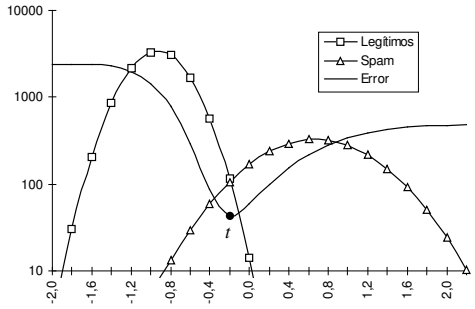
$$E_L(t) = N_L P_{L\to S}(t)$$

Y el número esperado de mensajes spam mal clasificados es

$$E_S(t) = N_S P_{S \to L}(t)$$

El error total esperado y ponderado con el coeficiente λ (lambda)

$$E(t) = \lambda E_L(t) + E_S(t)$$



Nuestro objetivo es minimizar esta ecuación. Para ello buscaremos el valor de *t* que minimice el error. Este valor cumple con las siguientes ecuaciones

$$E(t)\frac{d}{dt} = 0$$

$$E(t)\frac{d^2}{dt} > 0$$

Si calculamos la derivada en primer grado de la función de error tenemos

$$E(t)\frac{d}{dt} = N_S f_S(t) - \lambda N_L f_L(t)$$

O sea que debemos resolver esta ecuación

$$N_S f_S(t) - \lambda N_L f_L(t) = 0$$

O equivalentemente

$$N_S f_S(t) = \lambda N_L f_L(t)$$

Si expandimos la formula de la distribución normal

$$N_S \frac{e^{-(t-\mu_S)^2/2\sigma_S^2}}{\sigma_S\sqrt{2\pi}} = \lambda N_L \frac{e^{-(t-\mu_L)^2/2\sigma_L^2}}{\sigma_L\sqrt{2\pi}}$$

Haciendo un pasaje de términos y simplificando un poco

$$N_S \sigma_L e^{-(t-\mu_S)^2/2\sigma_S^2} = \lambda N_L \sigma_S e^{-(t-\mu_L)^2/2\sigma_L^2}$$

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Reagrupando los términos

$$\frac{e^{-(t-\mu_S)^2/2\sigma_S^2}}{e^{-(t-\mu_L)^2/2\sigma_L^2}} = \frac{\lambda N_L \sigma_S}{N_S \sigma_L}$$

O sea que

$$e^{(t-\mu_L)^2/2\sigma_L^2-(t-\mu_S)^2/2\sigma_S^2} = \frac{\lambda N_L \sigma_S}{N_S \sigma_L}$$

Aplicamos logaritmo

$$\frac{(t-\mu_L)^2}{2\sigma_L^2} - \frac{(t-\mu_S)^2}{2\sigma_S^2} = Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

Si multiplicamos ambos lados de la igualdad por $2\sigma_L^2\sigma_S^2$

$$\sigma_S^2(t-\mu_L)^2 - \sigma_L^2(t-\mu_S)^2 = 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

Expandiendo los cuadrados obtenemos

$$\sigma_S^2(t^2 - 2t\mu_L + \mu_L^2) - \sigma_L^2(t^2 - 2t\mu_S + \mu_S^2) = 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

Reagrupando los términos nos queda

$$(\sigma_S^2 - \sigma_L^2)t^2 + 2(\sigma_L^2\mu_S - \sigma_S^2\mu_L)t + \sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 = 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

O sea que

$$(\sigma_S^2 - \sigma_L^2)t^2 + 2(\sigma_L^2\mu_S - \sigma_S^2\mu_L)t + \sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right) = 0$$

Esta es polinomio en *t* de grado dos[27] que se puede resolver con la clásica ecuación

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Siendo *a*, *b* y *c*

$$a = \sigma_S^2 - \sigma_L^2$$

[27] Siempre que $\sigma_S^2 - \sigma_L^2 \neq 0$. El caso contrario lo analizaremos más adelante

$$b = 2\left(\sigma_L^2\mu_S - \sigma_S^2\mu_L\right)$$

$$c = \sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

Substituyendo *a*, *b* y *c* resulta

$$\frac{2\left(\sigma_S^2\mu_L - \sigma_L^2\mu_S\right) \pm \sqrt{4\left(\sigma_S^2\mu_L - \sigma_L^2\mu_S\right)^2 - 4\left(\sigma_S^2 - \sigma_L^2\right)\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)}}{2\left(\sigma_S^2 - \sigma_L^2\right)}$$

Pero si trabajamos un poco podemos simplificar aun más esta ecuación. Sacamos el "4" dentro de la raíz como factor común

$$\frac{2\left(\sigma_S^2\mu_L - \sigma_L^2\mu_S\right) \pm 2\sqrt{\left(\sigma_S^2\mu_L - \sigma_L^2\mu_S\right)^2 - \left(\sigma_S^2 - \sigma_L^2\right)\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)}}{2\left(\sigma_S^2 - \sigma_L^2\right)}$$

Y ahora podemos simplificar

$$\frac{\sigma_S^2\mu_L - \sigma_L^2\mu_S \pm \sqrt{\left(\sigma_S^2\mu_L - \sigma_L^2\mu_S\right)^2 - \left(\sigma_S^2 - \sigma_L^2\right)\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)}}{\sigma_S^2 - \sigma_L^2}$$

Ahora vamos a trabajar con el término dentro de la raíz cuadrada, así que para facilitar el seguimiento sólo transcribiremos dicha parte de la ecuación

$$\left(\sigma_S^2\mu_L - \sigma_L^2\mu_S\right)^2 - \left(\sigma_S^2 - \sigma_L^2\right)\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)$$

Si expandimos el cuadrado

$$\sigma_S^4\mu_L^2 - 2\sigma_S^2\mu_L\sigma_L^2\mu_S + \sigma_L^4\mu_S^2 + \left(\sigma_L^2 - \sigma_S^2\right)\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)$$

Si distribuimos los multiplicandos

$$\sigma_S^4\mu_L^2 - 2\sigma_S^2\mu_L\sigma_L^2\mu_S + \sigma_L^4\mu_S^2 + \sigma_L^2\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)$$
$$- \sigma_S^2\left(\sigma_S^2\mu_L^2 - \sigma_L^2\mu_S^2 - 2\sigma_L^2\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)$$

Si volvemos a distribuir

$$\sigma_S^4\mu_L^2 - 2\sigma_S^2\mu_L\sigma_L^2\mu_S + \sigma_L^4\mu_S^2 + \sigma_L^2\sigma_S^2\mu_L^2 - \sigma_L^4\mu_S^2 - 2\sigma_L^4\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$
$$- \sigma_S^4\mu_L^2 + \sigma_L^2\sigma_S^2\mu_S^2 + 2\sigma_L^2\sigma_S^4 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

Ahora podemos cancelar algunos sumandos, y nos queda

$$-2\sigma_S^2\mu_L\sigma_L^2\mu_S + \sigma_L^2\sigma_S^2\mu_L^2 - 2\sigma_L^4\sigma_S^2 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right) + \sigma_L^2\sigma_S^2\mu_S^2 + 2\sigma_L^2\sigma_S^4 Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Si sacamos factor común

$$\sigma_L^2\sigma_S^2\left(-2\mu_L\mu_S+\mu_L^2-2\sigma_L^2 Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)+\mu_S^2+2\sigma_S^2 Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)\right)$$

Factorizando

$$\sigma_L^2\sigma_S^2\left(\left(\mu_L-\mu_S\right)^2-2\sigma_L^2 Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)+2\sigma_S^2 Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)\right)$$

Y sacamos nuevamente factor común

$$\sigma_L^2\sigma_S^2\left(\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)\right)$$

Si reemplazamos esta ecuación en la fórmula original tenemos

$$\frac{\mu_L\sigma_S^2-\mu_S\sigma_L^2\pm\sqrt{\sigma_L^2\sigma_S^2\left(\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)\right)}}{\sigma_S^2-\sigma_L^2}$$

Y ahora podemos sacar de dentro de la raíz cuadrada $\sigma_L^2\sigma_S^2$

$$t=\frac{\mu_L\sigma_S^2-\mu_S\sigma_L^2\pm\sigma_L\sigma_S\sqrt{\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Aquí hallamos dos soluciones. A continuación demostraremos que la primera se trata de un mínimo y la segunda de un máximo.

Esta demostración es algo más larga. Y para evitar tener que repetirla por cada una de las soluciones, trabajaremos con ambas en forma simultánea.

La idea es para cada una de estas soluciones ver si se cumple esta desigualdad:

$$E(t)\frac{d^2}{dt}>0$$

O sea

$$\left(N_S\frac{e^{-(t-\mu_S)^2/2\sigma_S^2}}{\sigma_S\sqrt{2\pi}}-\lambda N_L\frac{e^{-(t-\mu_L)^2/2\sigma_L^2}}{\sigma_L\sqrt{2\pi}}\right)\frac{d}{dt}>0$$

Resolviendo la derivada

$$N_S\frac{\mu_S-t}{\sigma_S^3\sqrt{2\pi}}e^{-(t-\mu_S)^2/2\sigma_S^2}-\lambda N_L\frac{\mu_L-t}{\sigma_L^3\sqrt{2\pi}}e^{-(t-\mu_L)^2/2\sigma_L^2}>0$$

O lo que es lo mismo

$$N_S\frac{\mu_S-t}{\sigma_S^3\sqrt{2\pi}}e^{-(t-\mu_S)^2/2\sigma_S^2}>\lambda N_L\frac{\mu_L-t}{\sigma_L^3\sqrt{2\pi}}e^{-(t-\mu_L)^2/2\sigma_L^2}$$

Haciendo pasaje de términos

$$N_S\sigma_L^3(\mu_S - t)e^{-(t-\mu_S)^2/2\sigma_S^2} > \lambda N_L\sigma_S^3(\mu_L - t)e^{-(t-\mu_L)^2/2\sigma_L^2}$$

Agrupando las exponenciales

$$N_S\sigma_L^3(\mu_S - t)e^{(\mu_L-t)^2/2\sigma_L^2-(\mu_S-t)^2/2\sigma_S^2} > \lambda N_L\sigma_S^3(\mu_L - t)$$

Esta es la inecuación sobre la que vamos a trabajar. Pero para hacerlo más fácil de seguir lo haremos por partes. Primero resolveremos lo siguiente.

$$\mu_S - t$$

Reemplazando *t* por el umbral antes calculado queda

$$\mu_S - \frac{\mu_L\sigma_S^2 - \mu_S\sigma_L^2 \pm \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Lo que equivale a[28]

$$\frac{\mu_S(\sigma_S^2-\sigma_L^2) - \mu_L\sigma_S^2 + \mu_S\sigma_L^2 \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Distribuyendo μ_S

$$\frac{\mu_S\sigma_S^2 - \mu_S\sigma_L^2 - \mu_L\sigma_S^2 + \mu_S\sigma_L^2 \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Simplificando sumandos

$$\frac{\mu_S\sigma_S^2 - \mu_L\sigma_S^2 \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Sacando factor común

$$\frac{\sigma_S^2(\mu_S - \mu_L) \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

O sea que

$$\mu_S - t = \frac{\sigma_S^2(\mu_S-\mu_L) \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

[28] En este paso invertimos los signos ± por ∓ . Tener en cuenta que en toda esta demostración el signo superior siempre se corresponde con la primera solución (+) y el inferior con la segunda (-)

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Con un procedimiento equivalente resolvemos

$$\mu_L - t$$

$$\mu_L - \frac{\mu_L\sigma_S^2 - \mu_S\sigma_L^2 \pm \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

$$\frac{\mu_L(\sigma_S^2-\sigma_L^2) - \mu_L\sigma_S^2 + \mu_S\sigma_L^2 \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

$$\frac{\mu_L\sigma_S^2 - \mu_L\sigma_L^2 - \mu_L\sigma_S^2 + \mu_S\sigma_L^2 \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

$$\frac{-\mu_L\sigma_L^2 + \mu_S\sigma_L^2 \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

$$\frac{\sigma_L^2(\mu_S-\mu_L) \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

O sea que

$$\mu_L - t = \frac{\sigma_L^2(\mu_S-\mu_L) \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Volviendo a la inecuación original

$$N_S\sigma_L^3(\mu_S - t)e^{(\mu_L-t)^2/2\sigma_L^2 - (\mu_S-t)^2/2\sigma_S^2} > \lambda N_L\sigma_S^3(\mu_L - t)$$

Ahora vamos a trabajar en el exponente de *e*

$$\frac{(\mu_L-t)^2}{2\sigma_L^2} - \frac{(\mu_S-t)^2}{2\sigma_S^2}$$

Primero analizaremos el primer termino

$$\frac{(\mu_L-t)^2}{2\sigma_L^2}$$

Reemplazando el numerador de la división por el valor obtenido más arriba

$$\frac{\left(\frac{\sigma_L^2(\mu_S-\mu_L) \mp \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2 + 2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}\right)^2}{2\sigma_L^2}$$

Esto equivale a

$$\frac{\left(\sigma_L^2(\mu_S-\mu_L)\mp\sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2\sigma_L^2(\sigma_S^2-\sigma_L^2)^2}$$

En el dividendo podemos sacar factor común σ_L

$$\frac{\sigma_L^2\left(\sigma_L(\mu_S-\mu_L)\mp\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2\sigma_L^2(\sigma_S^2-\sigma_L^2)^2}$$

Y podemos simplificar σ_L^2

$$\frac{\left(\sigma_L(\mu_S-\mu_L)\mp\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2(\sigma_S^2-\sigma_L^2)^2}$$

O sea que

$$\frac{(\mu_L-t)^2}{2\sigma_L^2}=\frac{\left(\sigma_L(\mu_S-\mu_L)\mp\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2(\sigma_S^2-\sigma_L^2)^2}$$

Con un procedimiento similar podemos ver a que equivale

$$\frac{(\mu_S-t)^2}{2\sigma_S^2}$$

Veamos

$$\frac{\left(\frac{\sigma_S^2(\mu_S-\mu_L)\mp\sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}\right)^2}{2\sigma_S^2}$$

$$\frac{\left(\sigma_S^2(\mu_S-\mu_L)\mp\sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2\sigma_S^2(\sigma_S^2-\sigma_L^2)^2}$$

$$\frac{\sigma_S^2\left(\sigma_S(\mu_S-\mu_L)\mp\sigma_L\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2\sigma_S^2(\sigma_S^2-\sigma_L^2)^2}$$

$$\frac{\left(\sigma_S(\mu_S-\mu_L)\mp\sigma_L\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}\right)^2}{2(\sigma_S^2-\sigma_L^2)^2}$$

O sea que

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

$$\frac{(\mu_S - t)^2}{2\sigma_S^2} = \frac{\left(\sigma_S(\mu_S - \mu_L) \mp \sigma_L\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2}{2(\sigma_S^2 - \sigma_L^2)^2}$$

Ahora volviendo[29] al exponente de *e*

$$\frac{(\mu_L - t)^2}{2\sigma_L^2} - \frac{(\mu_S - t)^2}{2\sigma_S^2}$$

Y reemplazando por los valores obtenidos en los pasos anteriores

$$\frac{\left(\sigma_L(\mu_S - \mu_L) \mp \sigma_S\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2}{2(\sigma_S^2 - \sigma_L^2)^2} - \frac{\left(\sigma_S(\mu_S - \mu_L) \mp \sigma_L\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2}{2(\sigma_S^2 - \sigma_L^2)^2}$$

Agrupando

$$\frac{\left(\sigma_L(\mu_S - \mu_L) \mp \sigma_S\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2 - \left(\sigma_S(\mu_S - \mu_L) \mp \sigma_L\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2}{2(\sigma_S^2 - \sigma_L^2)^2}$$

Ahora expandimos ambos cuadrados y simplificamos algunos términos con signos inversos[30]

$$\frac{(\sigma_L(\mu_S - \mu_L))^2 + \left(\mp\sigma_S\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2 - (\sigma_S(\mu_S - \mu_L))^2 - \left(\mp\sigma_L\sqrt{(\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}\right)^2}{2(\sigma_S^2 - \sigma_L^2)^2}$$

Si aplicamos los cuadrados dentro de las expresiones en paréntesis

$$\frac{\sigma_L^2(\mu_S - \mu_L)^2 + \sigma_S^2\left((\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right) - \sigma_S^2(\mu_S - \mu_L)^2 - \sigma_L^2\left((\mu_L - \mu_S)^2 + 2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)\right)}{2(\sigma_S^2 - \sigma_L^2)^2}$$

Si distribuimos sobre la suma

$$\frac{\sigma_L^2(\mu_S - \mu_L)^2 + \sigma_S^2(\mu_L - \mu_S)^2 + 2\sigma_S^2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right) - \sigma_S^2(\mu_S - \mu_L)^2 - \sigma_L^2(\mu_L - \mu_S)^2 - 2\sigma_L^2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}{2(\sigma_S^2 - \sigma_L^2)^2}$$

Ahora hay algunos sumandos que se simplifican entre sí

$$\frac{2\sigma_S^2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right) - 2\sigma_L^2(\sigma_S^2 - \sigma_L^2)Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)}{2(\sigma_S^2 - \sigma_L^2)^2}$$

[29] Si se olvido que estabamos haciendo, le recuerdo que estamos tratando de simplificar el exponente de *e* en la ecuación original. Y esta ecuación nos servia para ver de los dos resultados obtenidos si correspondían a mínimos o máximos de la función de error

[30] Estimado lector, no desespere. Le prometemos que éste es el punto más complicado de la demostración a partir de ahora todo se va simplificando y llegaremos a una solución simple y elegante.

Si sacamos factor común

$$\frac{2\left(\sigma_S^2-\sigma_L^2\right)\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}{2\left(\sigma_S^2-\sigma_L^2\right)^2}$$

Lo que equivale a

$$\frac{2\left(\sigma_S^2-\sigma_L^2\right)^2 Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}{2\left(\sigma_S^2-\sigma_L^2\right)^2}$$

Ahora podemos simplificar

$$Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)$$

O sea que

$$\frac{\left(\mu_L-t\right)^2}{2\sigma_L^2}-\frac{\left(\mu_S-t\right)^2}{2\sigma_S^2}=Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)$$

Y volviendo a la inecuación original

$$N_S\sigma_L^3\left(\mu_S-t\right)e^{\left(\mu_L-t\right)^2/2\sigma_L^2-\left(\mu_S-t\right)^2/2\sigma_S^2}>\lambda N_L\sigma_S^3\left(\mu_L-t\right)$$

Y remplazando el resultado obtenido

$$N_S\sigma_L^3\left(\mu_S-t\right)e^{Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}>\lambda N_L\sigma_S^3\left(\mu_L-t\right)$$

Esto equivale a

$$N_S\sigma_L^3\left(\mu_S-t\right)\frac{\lambda N_L\sigma_S}{N_S\sigma_L}>\lambda N_L\sigma_S^3\left(\mu_L-t\right)$$

Haciendo algunas simplificaciones

$$\sigma_L^2\left(\mu_S-t\right)>\sigma_S^2\left(\mu_L-t\right)$$

Y podemos reemplazar los valores entre paréntesis por los resultados obtenidos anteriormente

$$\sigma_L^2\left(\frac{\sigma_S^2\left(\mu_S-\mu_L\right)\mp\sigma_L\sigma_S\sqrt{\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}\right)>\sigma_S^2\left(\frac{\sigma_L^2\left(\mu_S-\mu_L\right)\mp\sigma_L\sigma_S\sqrt{\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}\right)$$

Distribuyendo la multiplicación dentro del paréntesis

$$\frac{\sigma_L^2\sigma_S^2\left(\mu_S-\mu_L\right)\mp\sigma_L^3\sigma_S\sqrt{\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}>\frac{\sigma_S^2\sigma_L^2\left(\mu_S-\mu_L\right)\mp\sigma_L\sigma_S^3\sqrt{\left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Podemos dividir de ambos lados por $\sigma_L\sigma_S$

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

$$\frac{\sigma_L\sigma_S(\mu_S-\mu_L)\mp\sigma_L^2\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2} > \frac{\sigma_L\sigma_S(\mu_S-\mu_L)\mp\sigma_S^2\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Podemos también simplificar términos comunes

$$\frac{\mp\sigma_L^2\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2} > \frac{\mp\sigma_S^2\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

También podemos eliminar la raíz cuadrada

$$\frac{\mp\sigma_L^2}{\sigma_S^2-\sigma_L^2} > \frac{\mp\sigma_S^2}{\sigma_S^2-\sigma_L^2}$$

O lo que es lo mismo

$$\frac{\mp\sigma_L^2}{\sigma_S^2-\sigma_L^2} - \frac{\mp\sigma_S^2}{\sigma_S^2-\sigma_L^2} > 0$$

Agrupando la resta

$$\frac{\mp\sigma_L^2 \pm \sigma_S^2}{\sigma_S^2-\sigma_L^2} > 0$$

Lo que es igual a

$$\pm\frac{\sigma_S^2-\sigma_L^2}{\sigma_S^2-\sigma_L^2} > 0$$

Y finalmente simplificando

$$\pm 1 > 0$$

¿Qué quiere decir este resultado? Que si tomamos la primera solución de *t* es cierto que se trata de un mínimo de la función. Y si tomamos el segundo valor de *t*, este hace referencia a un máximo.

Con lo que concluimos que

$$t = \frac{\mu_L\sigma_S^2 - \mu_S\sigma_L^2 + \sigma_L\sigma_S\sqrt{(\mu_L-\mu_S)^2+2(\sigma_S^2-\sigma_L^2)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)}}{\sigma_S^2-\sigma_L^2}$$

Es el umbral óptimo del clasificador.

Pero para que este umbral sea "calculable" con esta fórmula se deben cumplir algunas condiciones.

1) $\sigma_S^2 - \sigma_L^2 \neq 0$

$$2)\ \left(\mu_L-\mu_S\right)^2+2\left(\sigma_S^2-\sigma_L^2\right)Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)\geq 0$$

Si la primera condición no se cumple[31], entonces

$$\sigma_S^2-\sigma_L^2=0$$

O sea que

$$\sigma_S^2=\sigma_L^2$$

Lo que quiere decir que

$$\sigma_S=\sigma_L$$

Y la ecuación para buscar el error mínimo

$$\left(\sigma_S^2-\sigma_L^2\right)t^2+2\left(\sigma_L^2\mu_S-\sigma_S^2\mu_L\right)t+\sigma_S^2\mu_L^2-\sigma_L^2\mu_S^2-2\sigma_L^2\sigma_S^2Log\left(\frac{\lambda N_L\sigma_S}{N_S\sigma_L}\right)=0$$

Se puede simplificar de la siguiente manera[32]

$$2\left(\sigma^2\mu_S-\sigma^2\mu_L\right)t+\sigma^2\mu_L^2-\sigma^2\mu_S^2-2\sigma^4Log\left(\frac{\lambda N_L}{N_S}\right)=0$$

Si dividimos ambos lados de la ecuación por σ^2

$$2\left(\mu_S-\mu_L\right)t+\mu_L^2-\mu_S^2-2\sigma^2Log\left(\frac{\lambda N_L}{N_S}\right)=0$$

Si despejamos[33] t

$$t=\frac{\mu_S^2-\mu_L^2+2\sigma^2Log\left(\frac{\lambda N_L}{N_S}\right)}{2\left(\mu_S-\mu_L\right)}$$

Y si distribuimos la división

$$t=\frac{\mu_S+\mu_L}{2}+\frac{\sigma^2}{\mu_S-\mu_L}Log\left(\frac{\lambda N_L}{N_S}\right)$$

A continuación veremos si esta solución que se trata de un punto mínimo del error corroborando que se cumple esta inecuación

$$E(t)\frac{d^2}{dt}>0$$

[31] Es bastante improbable que ocurra esto. Es prácticamente imposible que el desvío estándar de los mensajes spam coincida con el de los legítimos

[32] Como $\sigma_S=\sigma_L$ entonces para simplificar directamente lo llamamos σ

[33] Este paso se puede hacer si asumimos que $\mu_S\neq\mu_L$. De no cumplirse veremos que pasa más adelante

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

O sea

$$N_S \frac{\mu_S - t}{\sigma^3\sqrt{2\pi}} e^{-(t-\mu_S)^2/2\sigma^2} - \lambda N_L \frac{\mu_L - t}{\sigma^3\sqrt{2\pi}} e^{-(t-\mu_L)^2/2\sigma^2} > 0$$

O equivalentemente

$$N_S \frac{\mu_S - t}{\sigma^3\sqrt{2\pi}} e^{-(t-\mu_S)^2/2\sigma^2} > \lambda N_L \frac{\mu_L - t}{\sigma^3\sqrt{2\pi}} e^{-(t-\mu_L)^2/2\sigma^2}$$

Simplificando un poco

$$N_S(\mu_S - t)e^{-(t-\mu_S)^2/2\sigma^2} > \lambda N_L(\mu_L - t)e^{-(t-\mu_L)^2/2\sigma^2}$$

Y agrupando las exponenciales

$$N_S(\mu_S - t)e^{(t-\mu_L)^2/2\sigma^2 - (t-\mu_S)^2/2\sigma^2} > \lambda N_L(\mu_L - t)$$

Igual de lo que hicimos antes, atacaremos primero el coeficiente de la exponencial

$$\frac{(t-\mu_L)^2}{2\sigma^2} - \frac{(t-\mu_S)^2}{2\sigma^2}$$

Expandiendo los cuadrados

$$\frac{\mu_L^2 - 2t\mu_L + 2t\mu_S - \mu_S^2}{2\sigma^2}$$

Sacando factor común

$$\frac{\mu_L^2 - 2t(\mu_L - \mu_S) - \mu_S^2}{2\sigma^2}$$

Ahora reemplazamos *t* por el valor calculado antes

$$\frac{\mu_L^2 - 2\left(\frac{\mu_S + \mu_L}{2} + \frac{\sigma^2}{\mu_S - \mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)\right)(\mu_L - \mu_S) - \mu_S^2}{2\sigma^2}$$

Distribuimos el 2 dentro del paréntesis

$$\frac{\mu_L^2 - \left(\mu_S + \mu_L + \frac{2\sigma^2}{\mu_S - \mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)\right)(\mu_L - \mu_S) - \mu_S^2}{2\sigma^2}$$

Distribuimos la multiplicación sobre la resta

$$\frac{\mu_L^2 - \left(\mu_S + \mu_L + \frac{2\sigma^2}{\mu_S - \mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)\right)\mu_L + \left(\mu_S + \mu_L + \frac{2\sigma^2}{\mu_S - \mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)\right)\mu_S - \mu_S^2}{2\sigma^2}$$

Y distribuimos la multiplicación dentro del paréntesis

$$\frac{\mu_L^2 - \mu_L\mu_S - \mu_L^2 - \frac{2\sigma^2\mu_L}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right) + \mu_S^2 + \mu_L\mu_S + \frac{2\sigma^2\mu_S}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right) - \mu_S^2}{2\sigma^2}$$

Algunos términos se simplifican entre sí

$$\frac{\frac{2\sigma^2\mu_S}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right) - \frac{2\sigma^2\mu_L}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)}{2\sigma^2}$$

Ahora podemos simplificar la división

$$\frac{\mu_S}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right) - \frac{\mu_L}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)$$

Sacamos factor común

$$\frac{\mu_S-\mu_L}{\mu_S-\mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)$$

Simplificamos la división y llegamos a

$$Log\left(\frac{\lambda N_L}{N_S}\right)$$

O sea que

$$\frac{(t-\mu_L)^2}{2\sigma^2} - \frac{(t-\mu_S)^2}{2\sigma^2} = Log\left(\frac{\lambda N_L}{N_S}\right)$$

Por lo que la ecuación

$$N_S(\mu_S - t)e^{(t-\mu_L)^2/2\sigma^2 - (t-\mu_S)^2/2\sigma^2} > \lambda N_L(\mu_L - t)$$

Equivale a

$$N_S(\mu_S - t)e^{Log\left(\frac{\lambda N_L}{N_S}\right)} > \lambda N_L(\mu_L - t)$$

Con lo que desaparece la exponencial

$$N_S(\mu_S - t)\frac{\lambda N_L}{N_S} > \lambda N_L(\mu_L - t)$$

Simplificando un poco

$$\mu_S - t > \mu_L - t$$

Y sumando *t* de ambos lados

$$\mu_S > \mu_L$$

O sea que la ecuación original se trata de un mínimo solo sí el valor medio de los mensajes Spam es mayor que el de los Legítimos.

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

Con todo esto llegamos a este resultado final

$$t = \begin{cases} Si\ \sigma_S \neq \sigma_L & \begin{cases} Si\,\Delta \geq 0 & \frac{a+b\sqrt{\Delta}}{c} \\ Sino & +\infty \end{cases} \\ Sino & \begin{cases} Si\,\mu_S > \mu_L & d \\ Sino & +\infty \end{cases} \end{cases}$$

Siendo

$$\Delta = \left(\mu_L - \mu_S\right)^2 + 2\,c\,Log\left(\frac{\lambda N_L \sigma_S}{N_S \sigma_L}\right)$$

$$a = \mu_L \sigma_S^2 - \mu_S \sigma_L^2$$

$$b = \sigma_S \sigma_L$$

$$c = \sigma_S^2 - \sigma_L^2$$

$$d = \frac{\mu_S + \mu_L}{2} + \frac{\sigma^2}{\mu_S - \mu_L} Log\left(\frac{\lambda N_L}{N_S}\right)$$

Bibliografía

[AMA/97] GIANNI AMATI, FABIO CRESTANI, FLAVIO UBALDINI. A Learning System for Selective Dissemination of Information. Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence. pp.764-769. 1997

[AMA/99] GIANNI AMATIA, FABIO CRESTANI. Probabilistic Learning for Selective Dissemination of Information. Information Processing and Management n.35 pp.633-654. 1999

[AND/00a] ION ANDROUTSOPOULOS, JOHN KOUTSIAS, KONSTANTINOS V. CHANDRINOS AND CONSTANTINE D. SPYROPOULOS. An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Atenas, Grecia. pp.160-167. 2000

[AND/00b] ION ANDROUTSOPOULOS, JOHN KOUTSIAS, KONSTANTINOS V. CHANDRINOS, GEORGE PALIOURAS, CONSTANTINE D. SPYROPOULOS. An Evaluation of Naive Bayesian Anti-Spam Filtering. Proceedings of the workshop on Machine Learning in the New Information Age, European Conference on Machine Learning. Barcelona, España. pp.9-17. 2000

[AND/00c] ION ANDROUTSOPOULOS, GEORGIOS PALIOURAS, VANGELIS KARKALETSIS, GEORGIOS SAKKIS, CONSTANTINE D. SPYROPOULOS, PANAGIOTIS STAMATOPOULOS. Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. Proceedings of the workshop "Machine Learning and Textual Information Access", 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000). Lyon, Francia. pp. 1-13. 2000

[ARA/00] AVI ARAMPATZIS, JEAN BENEY, C.H.A. KOSTER, TH.P. VAN DER WEIDE. Incrementality, Half-life, and Threshold Optimization for Adaptive Document Filtering. In Proceeding of 9th Text Retrieval Conference (TREC-9). National Institute of Standards and Technology. pp.589-600. 2000

[ARA/01b] AVI ARAMPATZIS, ANDRÉ VAN HAMERAN. The Score-Distributional Threshold Optimization for Adaptive Binary Classification Tasks. Pro-

ceedings of the 24th annual international ACM SIGIR conference on Research and Development in Information Retrieval. pp.285-293. 2001

[ARA/01b] AVI ARAMPATZIS. Unbiased S-D Threshold Optimization, Initial Query Degradation, Decay, and Incrementality, for Adaptive Document Filtering. In Proceeding of 10th Text Retrieval Conference (TREC-10). National Institute of Standards and Technology. pp.596-603. 2001

[CAR/01] XAVIER CARRERAS, LLUIS MARQUEZ. Boosting Trees for Anti-Spam Email Filtering. Proceedings of RANLP-01, 3rd International Conference on Recent Advances in Natural Language Processing, pp.58-64, Bulgaria. 2001

[COA/60] RONALD COASE. The Problem of Social Cost. Journal of Law & Economics, vol. 2, pp.1-44. University of Chicago Press. 1960

[COH/96a] WILLIAM W. COHEN, YORAM SINGER. Context-sensitive learning methods for text categorization. Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval. pp.307-315. 1996

[COH/96b] WILLIAM W. COHEN. Learning Rules that Classify E-Mail. In Proceedings from the AAAI Spring Symposium on Machine Learning in Information Access, pp.18-25, 1996.

[COV/91] THOMAS M. COVER, JOY A. THOMAS. Elements of Information Theory. Publisher Wiley-Interscience. ISBN: 0471062596. August 12, 1991

[DON/79] JACK J. DONGARRA, CLEVE B. MOLER, JAMES R. BUNCH, G. W. STEWART, G.W. STEWART. LINPACK Users' Guide. Society for Industrial & Applied Mathematics. ISBN: 089871172X. 1979

[FUH/91] NORBERT FUHR, CHRIS BUCKLEY. A Probabilistic Learning Approach for Document Indexing. ACM Transactions on Information Systems, vol.9, no.3, pp.223-248. 1991

[GAL/00] LUIGI GALAVOTTI, FABRIZIO SEBASTIANI, MARIA SIMI. Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization. In Proceedings of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries. pp.59-68. 2000

[HAL/98] ROBERT J. HALL. How to avoid unwanted email. Communications of the ACM, vol.41 no.3, pp.88-95. 1998

[IYE/00] RAJ D. IYER, DAVID D. LEWIS, ROBERT E. SCHAPIRE, YORAM SINGER, AMIT SINGHAL. Boosting for Document Routing. 9th ACM Interna-

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

tional Conference on Information and Knowledge Management, pp.70-77. 2000

[JOH/94] GEORGE H. JOHN, RON KOHAVI, KARL PFLEGER. Irrelevant Features and the Subset Selection Problem. Proceedings of the 11th International Conference on Machine Learning. pp.121-129. 1994

[KIM/00] YU-HWAN KIM, SHANG-YOON HAHN, BYOUNG-TAK ZHANG. Text Filtering by Boosting Naive Bayes Classifiers. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 168-175. ACM Press. 2000

[KOL/96] DAPHNE KOLLER, MEHRAN SAHAMI. Toward Optimal Feature Selection. Proceedings of the 13th International Conference on Machine Learning (ML), Bari, Italia, pp.284-292. 1996

[LAM/99] WAI LAM, MIGUEL RUIZ, PADMINI SRINIVASAN. Automatic Text Categorization and Its Application to Text Retrieval. IEEE Transactions on Knowledge and Data Engineering. vol.11 no.6 pp.865-879. 1999

[LEW/92] DAVID D. LEWIS. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, Denmark. pp.37-50. 1992

[LEW/94] DAVID D. LEWIS, MARC RINGUETTE. A Comparison of Two Learning Algorithms for Text Categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval pp.81-93, Las Vegas, US. 1994

[LEW/96] DAVID D. LEWIS, ROBERT E. SCHAPIRE, JAMES P. CALLAN, RON PAPKA. Training Algorithms for Linear Text Classifiers. Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval. pp.298-306. 1996

[LEW/98] DAVID D. LEWIS. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. Proceedings of ECML-98, 10th European Conference on Machine Learning, pp.4-15. 1998

[MCC/98] ANDREW MCCALLUM, KAMAL NIGAM. A Comparison of Event Models for Naive Bayes Text Classification. AAAI/ICML-98 Workshop on Learning for Text Categorization, pp.41-48. Technical Report WS-98-05. AAAI Press. 1998

[MLA/98] DUNJA MLADENIC. Feature Subset Selection in Text-Learning. 10th European Conference on Machine Learning ECML98. Springer, Berlin, pp.95-100. 1998

[NGH/97] HWEE TOU NG, WEI BOON GOH, KOK LEONG LOW. Feature selection, perceptron learning, and a usability case study for text categorization. Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval. pp.67-73. 1997

[NIG/99] KAMAL NIGAM, JOHN LAFFERTY, ANDREW MCCALLUM. Using Maximum Entropy for Text Classification. In Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering. pp.61-67. 1999

[PRE/93] WILLIAM H. PRESS, BRIAN P. FLANNERY, SAUL A. TEUKOLSKY, WILLIAM T. VETTERLING. Numerical Recipes in C : The Art of Scientific Computing. Cambridge University Press; ISBN: 0521431085; 2nd edition. 1993

[SAH/98] SAHAMI, MEHRAN, DUMAIS, SUSAN, HECKERMAN, DAVID, HORVITZ, ERIC. A Bayesian Approach to Filtering Junk E-Mail. Learning for Text Categorization: Papers from the 1998 Workshop. AAAI Technical Report WS-98-05. 1998

[SAK/01] GEORGIOS SAKKIS, ION ANDROUTSOPOULOS, GEORGIOS PALIOURAS, VANGELIS KARKALETSIS, CONSTANTINE D. SPYROPOULOS, PANAGIOTIS STAMATOPOULOS. Stacking classifiers for anti-spam filtering of e-mail. Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing. pp.44-50. 2001

[SEB/02] FABRIZIO SEBASTIANI. Machine Learning in Automated Text Categorization. ACM Computing Surveys, vol.34 no.1 pp.1-47. 2002

[TAU/00a] DANIEL R. TAURITZ, IDA G. SPRINKHUIZEN-KUYPER. Adaptive Information Filtering: evolutionary computation and *n*-gram representation. Proceedings of the 12th Belgium-Netherlands Artificial Intelligence Conference. pp.157-164. 2000

[TAU/00b] DANIEL R. TAURITZ, JOOST N. KOK, IDA G. SPRINKHUIZEN-KUYPER. Adaptive Information Filtering using Evolutionary Computation. Information Sciences. vol.122 no.2-4 pp.121-140. 2000

[YAN/92] YIMING YANG, CHRISTOPHER G. CHUTE. A Linear Least Squares Fit Mapping Method for Information Retrieval from Natural Language Texts. Proceedings of the 14th International Conference on Computational Linguistics (COLING 92). McGraw-Hill, New York, pp.447-453. 1992

Tesistas: Pablo Alvarez Alonso LU 503/87
Daniel Mario Vega LU 195/88
Directora: Dra. Ana Silvia Haedo

[YAN/94] YIMING YANG , CHRISTOPHER G. CHUTE. An Example-Based Mapping Method for Text Categorization and Retrieval. ACM Transactions on Information Systems (TOIS), vol.12 pp.252-277. 1994

[YAN/95] YIMING YANG. Noise Reduction in a Statistical Approach to Text Categorization. Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval pp.256-263. 1995

[YAN/97a] YIMING YANG, JAN O. PEDERSEN. A Comparative Study on Feature Selection in Text Categorization. Proceedings of ICML-97, 14th International Conference on Machine Learning pp.412-420. 1997

[YAN/97b] YIMING YANG. An Evaluation of Statistical Approaches to Text Categorization. Technical Report CMU-CS-97-127, School of Computer Science, Carnegie Mellon University. 1997

[YAN/99a] YIMING YANG. An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval, vol.1, no.1/2, pp.69-90. 1999

[YAN/99b] YIMING YANG, XIN LIU. A re-examination of text categorization methods. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval. pp.42-49. 1999

[ZHA/98] CHENGXIANG ZHAI, PETER JANSEN, EMILIA STOICA, NORBERT GROT, DAVID A. EVANS. Threshold Calibration in CLARIT Adaptive Filtering. In Proceeding of 7th Text Retrieval Conference (TREC-7), National Institute of Standards and Technology, pp.149-156. 1998

[ZHA/01a] YI ZHANG, JAMIE CALLAN. A Generative Model for Filtering Thresholds. Proceedings of the Workshop on Language Modeling and Information Retrieval. Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. pp.97-102. 2001

[ZHA/01b] YI ZHANG, JAMIE CALLAN. Maximum Likelihood Estimation for Filtering Thresholds. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp.294-302. 2001